

ACCESS + EXCEL : Classe de
gestions des images avec Gdi+



gdipius

Documentation

Module VBA de gestion d'image

par Thierry GASPERMENT (arkham46.developpez.com/)

Date de publication : 15/05/08

Dernière mise à jour : 26/03/09

Module de classe VBA pour faciliter l'utilisation de la librairie graphique gdipius.dll (GDI+)
Version actuelle : v0.9

I - Introduction.....	5
II - Remarques préalables.....	5
II-A - Remarques.....	5
II-B - Pré-requis.....	6
II-C - Compatibilité avec Excel.....	6
III - Les fonctions de création et de sauvegarde.....	6
III-A - OpenFile.....	6
III-B - CloseFile.....	6
III-C - CreateBitmap.....	6
III-D - SaveFile.....	7
III-E - SaveTransformFile.....	7
III-F - SaveJpegLossLess.....	7
IV - Autres fonctions de lecture / écriture de l'image.....	8
IV-A - GetPixels.....	8
IV-B - SetPixels.....	9
IV-C - SetBitmap.....	9
IV-D - GetBitmap.....	9
IV-E - SetImageArray.....	10
IV-F - GetImageArray.....	10
IV-G - LoadBitmapFromControl.....	10
IV-H - GetPictureDisp.....	11
V - Les fonctions de modification de l'image.....	11
V-A - ReplaceColor.....	11
V-B - FillColor.....	11
V-C - ApplyColorMatrix.....	12
V-D - Resize.....	12
V-E - Crop.....	13
V-F - Rotate.....	13
V-G - RotateFlip.....	13
VI - Les fonctions de dessin.....	14
VI-A - DrawPixel.....	14
VI-B - DrawLine.....	14
VI-C - DrawRectangle.....	14
VI-D - DrawEllipse.....	15
VI-E - DrawPolygon.....	16
VI-F - DrawText.....	16
VI-G - DrawImage.....	17
VI-H - DrawBitmap.....	18
VII - Les fonctions pour gérer la liste d'images.....	19
VII-A - ImageListAdd.....	19
VII-B - ImageListGetBitmap.....	19
VII-C - ImageListAddBitmap.....	19
VII-D - ImageListAddRect.....	20
VII-E - ImageListAddFromControl.....	20
VII-F - ImageListNew.....	20
VII-G - ImageListExists.....	20
VII-H - ImageListDel.....	20
VII-I - ImageListClone.....	21
VII-J - ImageListWidth.....	21
VII-K - ImageListHeight.....	21
VII-L - ImageListGetPictureDisp.....	21
VII-M - Fonctions de dessin / transformation / gif animés.....	21
VIII - Les fonctions de restitution de l'image à l'écran.....	22
VIII-A - RepaintControl.....	22
VIII-B - RepaintControlNoFormRepaint.....	22
VIII-C - FastRepaint.....	23
VIII-D - FastRepaintSetCorrection.....	23
VIII-E - FastRepaintSetClipControl.....	23

VIII-F - FastRepaintResetClip.....	24
IX - Les fonctions pour gérer les régions.....	24
IX-A - CreateRegionEllipse.....	24
IX-B - CreateRegionPolygon.....	24
IX-C - CreateRegionRect.....	24
IX-D - DeleteRegion.....	25
IX-E - DeleteAllRegions.....	25
IX-F - FrameRegion.....	25
IX-G - FillRegion.....	25
IX-H - HatchRegion.....	25
IX-I - GetRegionXY.....	26
IX-J - PointInRegion.....	26
IX-K - GetRegionRect.....	26
IX-L - RegionExists.....	27
IX-M - RegionAddRegion.....	27
IX-N - RegionCombine.....	27
IX-O - CreateRegionFromColor.....	27
IX-P - ScaleRegion.....	28
IX-Q - TranslateRegion.....	28
IX-R - RotateRegion.....	28
IX-S - RegionsIntersect.....	28
IX-T - RegionClone.....	28
IX-U - SetFormRegion.....	29
IX-V - ResetFormRegion.....	29
X - Sauvegarde d'images en mémoire.....	29
X-A - KeepImage.....	29
X-B - ResetImage.....	29
X-C - ImageExists.....	30
X-D - DeleteImage.....	30
X-E - DeleteAllImages.....	30
X-F - GetImageBitmap.....	30
XI - Gestion des données Exif.....	30
XI-A - GetExifData.....	30
XI-B - SetExifData.....	30
XI-C - GenerateThumbnail.....	31
XI-D - ImportExifData.....	31
XII - Gestion des Gif animés.....	31
XII-A - GifSetFrame.....	31
XII-B - GifGetFrameCount.....	31
XII-C - GifGetFrameDelay.....	31
XIII - Gestion des textures.....	31
XIII-A - TextureAddFromFile.....	32
XIII-B - TextureAddFromImage.....	32
XIII-C - TextureAddFromControl.....	32
XIII-D - TextureDel.....	32
XIII-E - TextureTranslate.....	32
XIII-F - TextureWidth.....	32
XIII-G - TextureHeight.....	33
XIV - Gestion des polices de caractères.....	33
XIV-A - FontPrivateAddFromFile.....	33
XIV-B - FontPrivateAddFromByte.....	33
XIV-C - FontsPrivateDelete.....	33
XIV-D - FontsEnumerate.....	33
XV - Les fonctions de conversion.....	34
XV-A - ImgToCtrlX et ImgToCtrlY.....	34
XV-B - CtrlToImgX et CtrlToImgY.....	34
XV-C - PointsToPixelsX et PointsToPixelsY.....	34
XV-D - PixelToPointsX et PixelToPointsY.....	34

XVI - Les autres fonctions utiles.....	35
XVI-A - GetPixel.....	35
XVI-B - LongToRGB.....	35
XVI-C - SetXPTheme.....	35
XVI-D - SetDoubleBufferXP.....	36
XVI-E - DragForm.....	36
XVI-F - FontSizeToPixel.....	36
XVI-G - GetUserFormHandle.....	37
XVI-H - GetClientHandle.....	37
XVII - Les propriétés.....	37
XVII-A - ImageWidth et ImageHeight.....	37
XVII-B - ImageFormat.....	37
XVII-C - RefControl.....	37
XVII-D - PenAlignMode.....	38
XVII-E - FastRepaintReplace.....	38
XVII-F - WrapX et WrapY.....	38
XVII-G - DrawSmooth.....	38
XVII-H - DrawWithoutTransparency.....	38
XVII-I - FillTexture.....	38
XVII-J - PenTexture.....	38
XVIII - Conclusion.....	39
XIX - Les téléchargements.....	39

I - Introduction

GDI et GDI+ sont des bibliothèques graphiques.

En d'autres termes ce sont des fonctions que l'on peut appeler depuis nos programmes en ayant pris soin auparavant de les déclarer : ce sont les fameuses API.

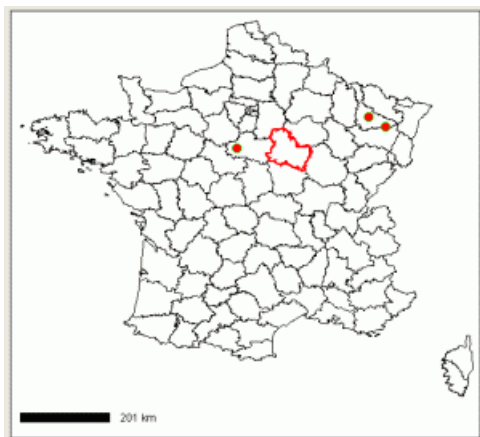
La bibliothèque GDI est intégrée à toutes les versions de Windows, elle contient des fonctions de dessins basiques et ne sait gérer que le format Bitmap (BMP)

Pour pallier le manque de fonctionnalités de la bibliothèque GDI, Microsoft a développé une autre bibliothèque : GDI+ (ou GdiPlus).

Cette seconde bibliothèque est beaucoup plus puissante et reconnaît en outre le format Jpeg et les données Exif intégrées aux images Jpeg, notamment par les appareils photos numériques.

La classe `clGdiPlus`, dont cet article est la documentation, encapsule ces API pour faciliter leur utilisations.

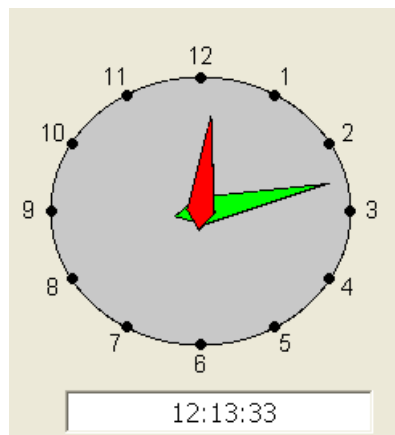
Exemples d'images obtenues grâce à cette classe :



La carte de France est dessinée à partir des coordonnées des départements.

Les départements sont encadrés au passage de la souris, et on peut les sélectionner en cliquant dessus.

Des cercles de couleur matérialisent l'emplacement de certaines villes.



Les aiguilles de l'horloge se déplacent en fonction de l'heure.

Si on tire sur une aiguille l'heure change.

II - Remarques préalables

II-A - Remarques

- L'**unité** utilisée par cette classe est le **pixel**.
- La classe est développée sur Access 2003 au format Access 2000; j'essaye autant que je peux d'assurer la meilleure compatibilité avec les autres versions.
- Dans tous mes exemples je nomme **clGdip** l'instance de la classe et **Image0** le contrôle image.
- Les **paramètres optionnels** sont notés : **[NomduParamètre] = ValeurParDéfaut**

- Si non mentionné, les **fonctions** renvoient **False** (Faux) en cas d'échec (ou **Vrai** si succès).

II-B - Pré-requis

Il est nécessaire de posséder la librairie GDI+ de Microsoft.

Cette librairie est incorporée à Windows XP.

La librairie est en téléchargement pour les systèmes d'exploitation suivants (Windows 2000; Windows 98; Windows ME; Windows NT; Windows XP; Windows Vista) :

Lien vers la librairie en téléchargement sur Microsoft.com

Le fichier gdiplus.dll doit être placé dans le répertoire de l'application.

II-C - Compatibilité avec Excel

Pour utiliser la classe CIGdiPlus avec Excel, recherchez la constante de compilation **Access** ; remplacez :

```
#Const Access = True
```

par

```
#Const Access = False
```

Le dessin se fait alors dans un contrôle sur un UserForm.

III - Les fonctions de création et de sauvegarde

III-A - OpenFile

Ouvre un fichier.

Charge l'image en mémoire.

On peut ensuite accéder aux données Exif ou modifier l'image.

La fonction renvoie **Vrai** si le fichier a pû être ouvert.

Paramètres :

Paramètre	Type	Explication
pFile	String	Chemin du fichier image
[pThumbNail]=False	Boolean	Mettre à True pour ouvrir la miniature Exif
[pIcon]=False	Boolean	Mettre à True pour ouvrir l'icone associée au fichier

III-B - CloseFile

Ferme le fichier (ou le bitmap créé avec CreateBitmap).

Pensez à refermer l'image dès que possible pour libérer de la mémoire.

Seul une image peut être ouverte à la fois : la fonction **CloseFile** n'a donc pas besoin de paramètre.

III-C - CreateBitmap

Création d'une image vierge.

L'image est créée en mémoire; on peut ensuite dessiner sur cette image.

Paramètres :

Paramètre	Type	Explication
pWidth	Long	Largeur de l'image
pHeight	Long	Hauteur de l'image
[pResolution] = 96	Single	Resolution de l'image (96 par défaut)

III-D - SaveFile

Sauvegarde l'image dans un fichier.

La fonction renvoie **Vrai** si le fichier a pû être sauvegardé.

Paramètres :

Paramètre	Type	Explication
pFile	String	Chemin du fichier de sauvegarde. Il n'est pas possible d'écraser le fichier courant.
pFormat	String	Format du fichier : JPG,GIF,BMP,PNG,TIF.
[pQuality]=-1	Integer	Qualité Jpeg (0-100) Ne spécifiez pas ce paramètre pour garder la qualité de l'image d'origine.

III-E - SaveTransformFile

Sauvegarde l'image dans un fichier avec transformation (sans perte Jpeg).

Paramètres :

Paramètre	Type	Explication
pFile	String	Chemin du fichier de sauvegarde. Il n'est pas possible d'écraser le fichier courant.
pTransformation	String	Transformation à appliquer

Exemple :

Rotation d'une image sans perte de qualité

```

If ClGdip.OpenFile("C:\MonFichier.jpg") then
  If ClGdip.SaveTransformFile("C:\MonFichierAvecRotation.jpg",EncoderValueTransformRotate90) then
    MsgBox "Image tournée et sauvegardée"
  End If
End If

```

III-F - SaveJpegLossLess

Sauvegarde l'image dans un fichier sans perte Jpeg.

A utiliser si uniquement modification de donnée Exif par exemple.

Paramètres :

Paramètre	Type	Explication
pFile	String	Chemin du fichier de sauvegarde. Il n'est pas possible d'écraser le fichier courant.

Exemple :

Function de changement de la date du cliché

```

' Change le tag DateTimeOriginal d'une image
Public Function ChangeImageDateTime(ByVal pPath As String, ByVal pDate As Date) As Boolean
    Dim clGdip As ClGdiPlus
    On Error GoTo Gestion_Erreurs
    ' Création nouvelle instance de classe
    Set clGdip = New ClGdiPlus
    ' Ouverture du fichier
    clGdip.OpenFile pPath
    ' Exif : Mise à jour de la date du cliché
    If clGdip.SetExifData(TagDateTimeOriginal, pDate) Then
        ' Sauvegarde de l'image dans un autre fichier
        If clGdip.SaveJpegLossLess(pPath & ".backup") Then
            ' Fermeture du fichier initial
            clGdip.CloseFile
            ' Supprime le fichier original
            Kill pPath
            ' Renomme l'image avec le nom d'origine
            Name pPath & ".backup" as pPath
            ChangeImageDateTime = True
        End If
    End If
    ' Libère l'instance de classe
    Set clGdip = Nothing
    On Error GoTo 0
Exit Function
Gestion_Erreurs:
    Set clGdip = Nothing
    ChangeImageDateTime = False
End Function
    
```

IV - Autres fonctions de lecture / écriture de l'image

IV-A - GetPixels

Renvoie un tableau de pixels avec les couleurs ARGB

Paramètres :

Paramètre	Type	Explication
pPixels	Tableau de Byte	Tableau de points ARGB

La fonction renvoie un tableau de pixels ARGB.

A = Alpha = Transparence

R = Red = Rouge

G = Green = Vert

B = Blue = Bleu

Les valeurs sont comprises entre 0 et 255.

Exemple :

Lire les couleurs des pixels

```

' Tableau pour recevoir les pixels
Dim lPixels() As Byte
' Compteurs
Dim lCptX As Long, lCptY As Long
' Lecture des pixels dans le tableau
lPixels = ClGdip.GetPixels
' Boucle sur les pixels
For lCptX = 1 To UBound(lPixels(), 2)
    For lCptY = 1 To UBound(lPixels(), 3)
    
```

Lire les couleurs des pixels

```
' Composante bleue = lPixels(1, lCptX, lCptY)
' Composante verte = lPixels(2, lCptX, lCptY)
' Composante rouge = lPixels(3, lCptX, lCptY)
' Composante alpha = lPixels(4, lCptX, lCptY)
```

Next

Next

IV-B - SetPixels

Redessine l'image à partir un tableau de pixels avec les couleurs ARGB

Paramètres :

Paramètre	Type	Explication
pPixels	Tableau de Byte	Tableau de points RGB

Exemple :

Appliquer un filtre rouge sur l'image

```
' Tableau pour recevoir les pixels
Dim lPixels() As Byte
' Compteurs
Dim lCptX As Long, lCptY As Long
' Lecture des pixels de l'image
lPixels = ClGdip.GetPixels
' On boucle sur les pixels
For lCptX = 1 To UBound(lPixels(), 2)
  For lCptY = 1 To UBound(lPixels(), 3)
    ' On ne garde que la couleur rouge en mettant les deux autres composantes à zéro
    ' Annule la composante bleue
    lPixels(1, lCptX, lCptY) = 0
    ' Annule la composante verte
    lPixels(2, lCptX, lCptY) = 0
    ' Composante bleue = lPixels(1, lCptX, lCptY)
    ' Composante verte = lPixels(2, lCptX, lCptY)
    ' Composante rouge = lPixels(3, lCptX, lCptY)
    ' Composante alpha = lPixels(4, lCptX, lCptY)
  Next
Next
' On réinjecte les couleurs dans l'image
ClGdip.SetPixels lPixels
```

IV-C - SetBitmap

Définit l'image à partir d'un bitmap gdipplus

Voir **GetBitmap** pour un exemple d'utilisation.

Paramètres :

Paramètre	Type	Explication
pBitmap	Long	Identifiant d'un bitmap GdiPlus
[pClone] = Vrai	Boolean	Si Vrai, clone le bitmap.

IV-D - GetBitmap

Sauvegarde l'image dans un bitmap gdipplus

Paramètres :

Paramètre	Type	Explication
[pClone] = Faux	Boolean	Si Vrai, clone le bitmap.

La fonction renvoie un bitmap GdiPlus.

Exemple :

Affecter l'image d'une classe à une autre

```
' On ne clone pas l'image lors de l'appel à GetBitmap.
' Par compte on clone l'image lors de l'appel à SetBitmap.
' Ainsi on obtient dans la classe ClGdiOut un clone de l'image issue de ClGdiIn
' Si on avait cloner deux fois l'image, l'image intermédiaire ne serait jamais libérée
' (=fuite de mémoire)
' Si on n'avait pas du tout cloner l'image, chaque classe pointerait vers la même image en mémoire
' (risque de libérer l'image dans une classe alors que l'autre voudrait encore l'utiliser)
lReturn = ClGdiOut.SetBitmap(ClGdiIn.GetBitmap(False), True)
```

IV-E - SetImageArray

Définit l'image à partir d'un tableau de byte

Voir l'application **Inclure des images jpeg dans un objet OLE** pour un exemple d'utilisation.

Paramètres :

Paramètre	Type	Explication
pArray	Tableau de Byte	Données de l'image.

IV-F - GetImageArray

Sauvegarde l'image dans un tableau de byte

Voir l'application **Inclure des images jpeg dans un objet OLE** pour un exemple d'utilisation.

Paramètres :

Paramètre	Type	Explication
pArray	Tableau de Byte	Données de l'image.
pFormat	String	Format du fichier : JPG,GIF,BMP,PNG,TIF.
[pQuality]=-1	Integer	Qualité Jpeg (0-100) Ne spécifiez pas ce paramètre pour garder la qualité de l'image d'origine.

La fonction retourne un tableau de Byte contenant les données brutes de l'image.

IV-G - LoadBitmapFromControl

Création d'un bitmap à partir de l'image contenue dans un contrôle

Paramètres :

Paramètre	Type	Explication
pCtrl	Image	Contrôle Image.

IV-H - GetPictureDisp

Renvoi l'image dans un objet IPictureDisp

Cet objet peut être utilisé pour une image du ruban.

V - Les fonctions de modification de l'image

V-A - ReplaceColor

Remplace une couleur de l'image par une autre.

Paramètres :

Paramètre	Type	Explication
pOldColor	Long	Ancienne couleur
pNewColor	Long	Nouvelle couleur
[pOldAlpha] = -1	Long	Ancienne transparence (0 --> 255) Si paramètre omis ou égal à -1, toutes les transparences de 0 à 255 sont traitées
[pNewAlpha] = -1	Long	Nouvelle transparence (0 --> 255) Si paramètre omis ou égal à -1, la transparence n'est pas modifiée
[pX1] = -1 [pY1] = -1 [pX2] = -1 [pY2] = -1	Long	Coordonnées du rectangle à traiter

Exemple :

Remplace le blanc par la couleur de fond du formulaire

```
lReturn = ClGdip.ReplaceColor(vbWhite, Me.Section(acDetail).BackColor)
```

V-B - FillColor

Rempli l'image d'une couleur.

Paramètres :

Paramètre	Type	Explication
pColor	Long	Couleur de remplissage
[pColorGradient]=-1	Long	Couleur pour dégradé Le dégradé va de la couleur pColor vers la couleur pColorGradient
[pGradientVert] = Faux	Boolean	Mettre à Vrai pour un dégradé vertical
[pX1] = -1 [pY1] = -1 [pX2] = -1 [pY2] = -1	Long	Coordonnées du rectangle à remplir.
[pAlpha] = 255	Integer	Transparence (0 --> 255)

Exemple :

Rempli de blanc le quart de l'image en bas à droite

Rempli de blanc le quart de l'image en bas à droite

```
lReturn = ClGdip.FillColor(vbWhite, , , ClGdip.ImageWidth / 2, ClGdip.ImageHeight / 2)
```

V-C - ApplyColorMatrix

Applique une matrice de couleurs à l'image.

Paramètres :

Paramètre	Type	Explication
pRed, pGreen, pBlue, pAlpha	Variante = Tableaux de single de dimension 5	1ère à 3ème valeur: Multiplicateurs des composantes rouge, vert, bleu 4ème valeur : Constante 5ème valeur : Valeur à ajouter Les valeurs doivent être comprises entre -1 et 1
[pEmptyColor] = 0	Long	Couleur de remplissage (-1 pour dessiner par dessus l'image). Paramètre utile si on utilise de la transparence.
[pX1] = -1 [pY1] = -1 [pX2] = -1 [pY2] = -1	Long	Coordonnées du rectangle à traiter.

Exemple :

Appliquer un filtre noir et blanc

```
ClGdip.ApplyColorMatrix Array(0.222, 0.707, 0.071, 0, 0), _
                        Array(0.222, 0.707, 0.071, 0, 0), _
                        Array(0.222, 0.707, 0.071, 0, 0), _
                        Array(0, 0, 0, 1, 0), -1
```

V-D - Resize

Redimensionne l'image.

Paramètres :

Paramètre	Type	Explication
pWidth	Long	Nouvelle largeur
pHeight	Long	Nouvelle hauteur
[pAntialias] = Faux	Boolean	Si vrai alors l'image est lissée
[pKeepExifData] = Faux	Boolean	Si Vrai alors les données Exif sont réinjectées dans l'image après redimensionnement

Exemple :

Redimensionne l'image à 10% de sa taille d'origine

```
lReturn = ClGdip.Resize(ClGdip.ImageWidth / 10, ClGdip.ImageHeight / 10)
```

V-E - Crop

Découpe l'image.

Paramètres :

Paramètre	Type	Explication
[pLeft] = 0	Long	Position à gauche
[pTop] = 0	Long	Position en haut
[pWidth] = 0	Long	Largeur
[pHeight] = 0	Long	Hauteur
[pAntialise] = Faux	Boolean	Si vrai alors l'image est lissée
[pKeepExifData] = Faux	Boolean	Si Vrai alors les données Exif sont réinjectées dans l'image après le découpage

Exemple :

Découpe le quart haut-gauche de l'image

```
lReturn = ClGdip.Crop(0, 0, ClGdip.ImageWidth / 2, ClGdip.ImageHeight / 2)
```

V-F - Rotate

Rotation de l'image.

Paramètres :

Paramètre	Type	Explication
pAngle	Long	Angle de rotation en degrés
[pKeepExifData] = Faux	Boolean	Si Vrai alors les données Exif sont réinjectées dans l'image après la rotation

Exemple :

Tourne l'image de 30°

```
lReturn = ClGdip.Rotate(30)
```

V-G - RotateFlip

Rotation/Miroir de l'image.

Paramètres :

Paramètre	Type	Explication
pType	Long	Type de transformation (énumération EGdipRotateFlip)
[pKeepExifData] = Faux	Boolean	Si Vrai alors les données Exif sont réinjectées dans l'image après la rotation et/ou le miroir

Exemple :

Retourne l'image horizontalement

```
lReturn = ClGdip.RotateFlip(RotateNoneFlipX)
```

Retourne l'image horizontalement

VI - Les fonctions de dessin

VI-A - DrawPixel

Dessine un pixel.

Paramètres :

Paramètre	Type	Explication
pX	Long	Position horizontale du pixel
pY	Long	Position verticale du pixel
pColor	Long	Couleur du pixel
[pAlpha] = 255	Integer	Transparence du pixel (0 --> 255)

Exemple :

Dessiner un pixel rouge au milieu du contrôle

```
lReturn = ClGdip.DrawPixel(cIgdip.ImageWidth,cIgdip.ImageHeight,vbRed)
```

VI-B - DrawLine

Dessine une ligne.

Paramètres :

Paramètre	Type	Explication
pX1,pY1	Long	Position du premier point
pX2,pY2	Long	Position du deuxième point
[pPenColor] = vbBlack	Long	Couleur de la ligne
[pPenWidth] = 1	Long	Épaisseur de la ligne
[pDash] = 0	Boolean	Pointillés si Vrai
[pAlpha] = 255	Integer	Transparence du trait (0 --> 255)
[pArrowLength] = 0	Long	Taille des traits de la flèche Si > 0, affiche une flèche
[pStartAndEnd] = 0	Long	Type de départ et de fin du trait (plat, carré, rond, triangle) Voir énumération EGdipLineCap

Exemple :

Dessiner une ligne bleue pointillée de 4 pixels de large qui traverse le contrôle en diagonal

```
lReturn = ClGdip.DrawLine(0, 0, ClGdip.ImageWidth, ClGdip.ImageHeight, vbBlue, 4, True)
```

VI-C - DrawRectangle

Dessine un rectangle.

Paramètres :

Paramètre	Type	Explication
pX1,pY1,pX2,pY2	Long	Coordonnées du rectangle
[pBackColor] = -1	Long	Couleur de remplissage Laisser -1 pour un rectangle transparent.
[pPenColor] = vbBlack	Long	Couleur de la ligne
[pPenWidth] = 1	Long	Épaisseur de la ligne
[pDash] = 0	Boolean	Pointillés si Vrai
[pAlpha] = 255	Integer	Transparence du trait (0 --> 255)
[pRegion] = ""	String	Nom de la région à créer avec les mêmes coordonnées

Exemple :

Dessiner un rectangle aléatoire

```
' pBackColor = -1 signifie qu'on dessine uniquement la bordure sans le fond
' pPenColor = vbBlue pour dessiner en bleu
lReturn = ClGdip.DrawRectangle(Rnd * ClGdip.ImageWidth, Rnd * ClGdip.ImageHeight, _
    Rnd * ClGdip.ImageWidth, Rnd * ClGdip.ImageHeight, _
    -1, vbBlue, 4)
```

VI-D - DrawEllipse

Dessine une ellipse.

Paramètres :

Paramètre	Type	Explication
pX1,pY1,pX2,pY2	Long	Position de l'ellipse
[pType] = 0	Integer	Type de positionnement Si pType = 0 : On passe un rectangle en paramètre, l'ellipse remplit ce rectangle pX1,Y1 : Point Haut-Gauche du rectangle pX2,Y2 : Point Bas-Droite du rectangle Si pType = 1 : on passe le centre et les rayons de l'ellipse en paramètre pX1,pY1 : Centre de l'ellipse pX2 : Rayon horizontal pY2 : Rayon vertical
[pBackColor] = -1	Long	Couleur de remplissage Laisser -1 pour une ellipse transparente
[pPenColor] = vbBlack	Long	Couleur de la ligne
[pPenWidth] = 1	Long	Épaisseur de la ligne
[pDash] = 0	Boolean	Pointillés si Vrai
[pAlpha] = 255	Integer	Transparence du trait (0 --> 255)
[pStartAngle] = Null	Single	Angle de démarrage d'un arc en degré
[pSweepAngle] = Null	Single	Angle de balayage d'un arc en degré
[pRegion] = ""	String	Nom de la région à créer avec les mêmes coordonnées

Exemple :

Dessiner une ellipse au milieu du contrôle

```
' Dessin d'une ellipse rouge d'épaisseur 2, centrée au milieu du contrôle, remplie de vert,
' et de rayons 1/5è de la taille du contrôle
lReturn = ClGdip.DrawEllipse _
```

Dessiner une ellipse au milieu du contrôle

```
(ClGdip.ImageWidth / 2, ClGdip.ImageHeight / 2, _
ClGdip.ImageWidth / 5, ClGdip.ImageHeight / 5, 1, _
vbGreen, vbRed, 2)
```

VI-E - DrawPolygon
Dessine un polygone.

Paramètres :

Paramètre	Type	Explication
pPoints	Variante	Tableau de points formant le polygone : Array(X1,Y1,X2,Y2,X3,Y3,...)
[pBackColor] = -1	Long	Couleur de remplissage Laisser -1 pour un rectangle transparent.
[pPenColor] = vbBlack	Long	Couleur de la ligne
[pPenWidth] = 1	Long	Epaisseur de la ligne
[pDash] = 0	Boolean	Pointillés si Vrai
[pAlpha] = 255	Integer	Transparence du trait (0 --> 255)
[pRegion] = ""	String	Nom de la région à créer avec les mêmes coordonnées

VI-F - DrawText
Dessine du texte.

Renvoie un boolean indiquant si le dessin s'est correctement déroulé.

Paramètres :

Paramètre	Type	Explication
pText	String	Texte à écrire Peut contenir des retours à la ligne (vbCrLf)
pSize	Long	Taille du texte en pixel
pFontName	String	Nom de la police (Arial, ...)
pX1,pY1,[pX2] = -1,[pY2] = -1	Long	Position du texte. Le texte est écrit dans le rectangle défini par ces coordonnées Si pX2 et pY2 ne sont pas renseignés, le texte est positionné sur le point de coordonnées pX1,pY1.
[pAlignHoriz] = 0	Integer	Alignement horizontal - 0 : Centre - 1 : Gauche - 2 : Droite
[pAlignVert] = 0	Integer	Alignement vertical - 0 : Centre - 1 : Haut

		- 2 : Bas
[pPenColor] = 0	Long	Couleur du texte
[pPenAlpha] = 255	Integer	Transparence du texte (0 --> 255)
[pBackColor] = -1	Long	Couleur de fond (-1 si transparent)
[pBackAlpha] = -1	Integer	Transparence du fond (0 --> 255)
[pItalic] = False	Boolean	Italique Vrai/Faux
[pBold] = False	Boolean	Gras Vrai/Faux
[pUnderline] = False	Boolean	Souligné Vrai/Faux
[pStrikeOut] = False	Boolean	Barré Vrai/Faux
[pAntialias] = Faux	Boolean	Lisser le texte Vrai/Faux
[pOnlyGetSize] = Faux	Boolean	Si vrai, n'écrit pas le texte et renvoie la taille du texte dans pX1,pY1,pX2,PY2

Exemple :

Ecrit un texte en haut à gauche de l'image, en bleu

```
lReturn = ClGdip.DrawText("TEST", 30, "Arial", 0, 0, ClGdip.ImageWidth,
ClGdip.ImageHeight, 0, 0, vbBlue)
```

VI-G - DrawImage

Affiche une image de la liste d'images.

Paramètres :

Paramètre	Type	Explication
pName	String	Identifiant de l'image
pX1,pY1,[pX2],[pY2]	Long	Position de l'image Si pX2 et pY2 sont nuls alors on conserve la taille de l'image Si pX2 = 0 ou pY2 =0 alors on calcule la dimension manquante par rapport à l'autre en conservant le rapport hauteur/largeur. Si pImgSizeMode = acOLESizeAutoSize : pX1,pY1 : Position de l'image pX2 : Largeur de l'image pY2 : Hauteur de l'image Si pImgSizeMode = acOLESizeZoom/ acOLESizeStretch/acOLESizeClip : pX1,Y1 : Point Haut-Gauche du rectangle contenant l'image pX2,Y2 : Point Bas-Droite du rectangle contenant l'image
[pTranspcolor] = -1	Integer	Couleur de transparence : les points de cette couleur ne

		seront pas dessinés
[pImgSizeMode] = acOLESizeStretch	Integer	Type d'affichage de l'image (Echelle par défaut)
[pImgPictureAlignment] = 2	Integer	Position de l'image (centrée par défaut)
[pPercent]=255	Integer	Pourcentage pour affichage translucide (0:invisible;255:normal)
[pAntialias] = Faux	Boolean	Lisse l'image si Vrai
[pRegion] = ""	String	Ajoute une région de nom pRegion
[pRegionColor] = -1	Long	Si renseigné, la région ajoutée contient les points de couleurs différentes de pRegionColor
[pRegionAlpha] = -1	Long	Si renseigné, la région ajoutée contient les points de transparences différentes de pRegionAlpha

VI-H - DrawBitmap

Dessine un bitmap gdiplus sur l'image.

Paramètres :

Paramètre	Type	Explication
pBitmap	Long	Identifiant du bitmap GdiPlus
pX1,pY1,[pX2],[pY2]	Long	Position de l'image Si pX2 et pY2 sont nuls alors on conserve la taille de l'image Si pX2 = 0 ou pY2 = 0 alors on calcule la dimension manquante par rapport à l'autre en conservant le rapport hauteur/largeur. Si pImgSizeMode = acOLESizeAutoSize : pX1,pY1 : Position de l'image pX2 : Largeur de l'image pY2 : Hauteur de l'image Si pImgSizeMode = acOLESizeZoom/ acOLESizeStretch/acOLESizeClip : pX1,Y1 : Point Haut-Gauche du rectangle contenant l'image pX2,Y2 : Point Bas-Droite du rectangle contenant l'image
[pTranspcolor] = -1	Integer	Couleur de transparence : les points de cette couleur ne seront pas dessinés
[pImgSizeMode] = acOLESizeStretch	Integer	Type d'affichage de l'image (Echelle par défaut)
[pImgPictureAlignment] = 2	Integer	Position de l'image (centrée par défaut)
[pPercent]=255	Integer	Pourcentage pour affichage translucide (0:invisible;255:normal)
[pAntialias] = Faux	Boolean	Lisse l'image si Vrai
[pRegion] = ""	String	Ajoute une région rectangulaire de nom pRegion
[pRegionColor] = -1	Long	Si renseigné, la région ajoutée contient les points de couleurs différentes de pRegionColor
[pRegionAlpha] = -1	Long	Si renseigné, la région ajoutée contient les points de transparences différentes de pRegionAlpha

VII - Les fonctions pour gérer la liste d'images

VII-A - ImageListAdd

Ajoute une image à la liste d'image.

Paramètres :

Paramètre	Type	Explication
pName	String	Identifiant de l'image
[pFile]=""	String	Chemin du fichier de l'image Si aucun fichier n'est passé en paramètre alors on sauvegarde l'image en mémoire dans la liste d'image
[pWidth]=0 [pHeight]=0	Long	Taille de l'image Si pWidth et pHeight sont nulles toutes les deux l'image conserve sa taille Si seulement une des deux tailles est nulles alors l'autre est calculée en conservant les proportions de l'image d'origine.
[pAntialias] = Faux	Boolean	Lisse l'image lors du redimensionnement si Vrai.
[pThumbNail] = Faux	Boolean	Lecture de la miniature intégrée à un Jpeg (Nécessite GDI+)
[plcon] = Faux	Boolean	Si Vrai, extrait l'icone associée au fichier.

VII-B - ImageListGetBitmap

Renvoie le bitmap gdiplus d'une image de la liste.

Voir `ImageListAddBitmap` pour un exemple d'utilisation.

Paramètres :

Paramètre	Type	Explication
pName	String	Identifiant de l'image
[pClone]= Faux	Boolean	Si Vrai, clone le bitmap lu dans la liste d'images.

VII-C - ImageListAddBitmap

Ajoute une image à la liste d'images à partir d'un bitmap gdiplus.

Paramètres :

Paramètre	Type	Explication
pName	String	Identifiant de l'image
pBitmap	Long	Identifiant d'un bitmap GdiPlus
[pClone]= True	Boolean	Si Vrai, clone le bitmap avant de l'insérer dans la liste d'images.

Exemple :

Passage d'une image d'une classe à un autre classe

```
' On ne clone pas l'image lors de l'appel à ImageListGetBitmap.
' Par compte on clone l'image lors de l'appel à ImageListAddBitmap.
' Ainsi on obtient dans la classe clGdiOut un clone de l'image issue de clGdiIn
' Si on avait cloner deux fois l'image, l'image intermédiaire ne serait jamais libérée
' (=fuite de mémoire)
```

Passage d'une image d'une classe à un autre classe

```
' Si on n'avait pas du tout cloner l'image, chaque classe pointerait vers la même image en mémoire
  ' (=>risque de libérer l'image dans une classe alors que l'autre voudrait encore l'utiliser)
lReturn = ClGdipOut.ImageListAddBitmap("MonImage",
ClGdipIn.ImageListGetBitmap("MonImage", False), True)
```

VII-D - ImageListAddRect

Ajoute une image à la liste d'images à partir d'une portion de l'image courante.

Paramètres :

Paramètre	Type	Explication
pName	String	Identifiant de l'image
pX1,pY1,pX2,pY2	Long	Coordonnées du rectangle à copier.

VII-E - ImageListAddFromControl

Ajoute une image à la liste d'images à partir de l'image intégrée à un contrôle.

Paramètres :

Paramètre	Type	Explication
pName	String	Identifiant de l'image
pCtrl	Image	Contrôle image contenant l'image à charger.

VII-F - ImageListNew

Crée une image vierge dans la liste d'images.

Paramètres :

Paramètre	Type	Explication
pName	String	Identifiant de l'image
pWidth	Long	Largeur de l'image
pHeight	Long	Hauteur de l'image

VII-G - ImageListExists

Teste si l'image existe déjà dans la liste.

La fonction renvoie Vrai si l'image est déjà dans la liste

Paramètres :

Paramètre	Type	Explication
pName	String	Identifiant de l'image

VII-H - ImageListDel

Supprime une image de la liste d'image.

Paramètres :

Paramètre	Type	Explication
pName	String	Identifiant de l'image

VII-I - ImageListClone

Clone une image de la liste d'image.

Paramètres :

Paramètre	Type	Explication
pNameSource	String	Identifiant de l'image source
pNameDest	String	Identifiant de l'image destination

VII-J - ImageListWidth

Largeur d'une image de la liste.

Paramètres :

Paramètre	Type	Explication
pName	String	Identifiant de l'image

VII-K - ImageListHeight

Hauteur d'une image de la liste.

Paramètres :

Paramètre	Type	Explication
pName	String	Identifiant de l'image

VII-L - ImageListGetIPictureDisp

Renvoie l'image dans un objet IPictureDisp

Cet objet peut être utilisé pour une image du ruban.

Paramètre	Type	Explication
pName	String	Identifiant de l'image

VII-M - Fonctions de dessin / transformation / gif animés

Les fonctions de dessin / transformation / gif animés existent également pour les images de la liste d'images.

Ces fonctions ont un préfixe **ImageList** et acceptent les même paramètres que les fonctions originales.

Le premier paramètre supplémentaire **PImage** est le nom de l'image de la liste.

Liste de ces fonctions:

Nom de la fonction	Description
ImageListDrawImage	Affiche une image de la liste d'images sur une autre image de la liste
ImageListDrawBitmap	Dessine un bitmap gdiplus sur une image de liste
ImageListDrawLine	Dessine une ligne sur une image de la liste
ImageListDrawPixel	Dessine un pixel sur une image de la liste
ImageListDrawRectangle	Dessine un rectangle sur une image de la liste
ImageListDrawEllipse	Dessine une ellipse sur une image de la liste
ImageListDrawPolygon	Dessine un polygone sur une image de la liste
ImageListDrawText	Dessine du texte sur une image de la liste
ImageListGetPixel	Renvoie la couleur d'un pixel d'une image de la liste
ImageListGetPixels	Renvoie un tableau de long contenant les pixels d'une image de la liste
ImageListSetPixels	Injecte un tableau de byte contenant les pixels dans une image de la liste
ImageListRotate	Rotation d'une image de la liste
ImageListRotateFlip	Rotation/Miroir d'une image de la liste
ImageListResize	Redimensionne une image de la liste
ImageListCrop	Découpe une image de la liste
ImageListCrop	Découpe une image de la liste
ImageListFillColor	Remplit une image de la liste d'une couleur
ImageListReplaceColor	Remplace une couleur d'une image de la liste
ImageListApplyColorMatrix	Applique une matrice de couleurs à une image de la liste
ImageListGifSetFrame	Change l'image courante d'un gif animé de la liste d'images
ImageListGifGetFrameCount	Nombre d'images d'un gif animé de la liste d'images
ImageListGifGetFrameDelay	Délais d'affichage des images d'un gif animé de la liste d'images

VIII - Les fonctions de restitution de l'image à l'écran

VIII-A - RepaintControl

Injecte l'image dans un contrôle.

Paramètres :

Paramètre	Type	Explication
pControl	Image	Contrôle dans lequel est injecté l'image
[pResizeBefore]=False	Boolean	Si Vrai, redimensionne l'image avant
[pResizeAntialias]=False	Boolean	Anti-aliasing lors du redimensionnement
[pUseEMF]=False	Boolean	Pour ACCESS uniquement : Utilisation du type d'image EMF A essayer pour améliorer la qualité de l'image dessinée.

Affichage de l'image dans le contrôle Image0 avec redimensionnement et lissage

```
clGdiplus.RepaintControl Me.Image0, True, True
```

VIII-B - RepaintControlNoFormRepaint

Injecte l'image dans un contrôle, mais ne la dessine que temporairement.

Cette fonction peut être utilisée pour réduire les scintillements (ou "flash") à l'écran.

Il peut y avoir des légers décalages d'un ou deux pixels, qu'on peut essayer de corriger avec la fonction **FastRepaintSetCorrection** et/ou en retirant les bordures de l'image.

Paramètres :

Paramètre	Type	Explication
pControl	Image	Contrôle dans lequel est injecté l'image
[pResizeBefore]=False	Boolean	Si Vrai, redimensionne l'image avant
[pResizeAntialias]=False	Boolean	Anti-aliasing lors du redimensionnement
[pUseEMF]=False	Boolean	Pour ACCESS uniquement : Utilisation du type d'image EMF A essayer pour améliorer la qualité de l'image dessinée.

VIII-C - FastRepaint

Dessine directement sur le formulaire.

L'affichage est plus rapide mais temporaire.

Il peut y avoir des légers décalages d'un ou deux pixels, qu'on peut essayer de corriger avec la fonction **FastRepaintSetCorrection** et/ou en retirant les bordures de l'image.

Paramètres :

Paramètre	Type	Explication
pImgCtrl	Image	Contrôle déterminant l'emplacement de l'image
[pTranspColor] = -1	Long	Couleur à ne pas dessiner.
[pPercent] = 255	Integer	Transparence (0 à 255).
[pAntialias] = Faux	Boolean	Lisse l'image si Vrai.
[pHdc] = 0	Long	Handle d'un device contexte si connu.
[pHwnd] = 0	Long	Handle d'une fenêtre si connu.

VIII-D - FastRepaintSetCorrection

Corrections pour la fonction FastRepaint.

*Dest pour correction sur le formulaire.

*Src pour correction sur l'image dessiné.

Paramètres :

Paramètre	Type	Explication
pLeftDest, pLeftSrc	Long	Correction à gauche
pTopDest, pTopSrc	Long	Correction à droite
pWidthDest, pWidthSrc	Long	Correction de largeur
pHeightDest, pHeightSrc	Long	Correction de hauteur

VIII-E - FastRepaintSetClipControl

Définit l'emplacement à redessiner avec la fonction FastRepaint.

Permet de placer des contrôles sur l'images et de ne pas dessiner par dessus si on utilise la fonction **FastRepaint**.

Paramètres :

Paramètre	Type	Explication
pCtrl	Contrôle	Contrôle dont l'emplacement est à inclure ou exclure
pInclude	Boolean	Si Vrai, le contrôle spécifié délimite un rectangle qui sera redessiné à l'appel de la fonction FastRepaint .

		Si Faux, le contrôle spécifié délimite un rectangle qui ne sera pas redessiné.
pCorrectLeft	Long	Correction à gauche en pixels
pCorrectTop	Long	Correction en haut en pixels
pCorrectRight	Long	Correction à droite en pixels
pCorrectBottom	Long	Correction en bas en pixels

VIII-F - FastRepaintResetClip

Réinitialise l'emplacement à redessiner avec le fonction FasRepaint.

Après l'appel de cette fonction, l'image entière est à nouveau dessinée.

IX - Les fonctions pour gérer les régions

IX-A - CreateRegionEllipse

Ajoute une région elliptique.

Paramètres :

Paramètre	Type	Explication
pRegion	String	Identifiant de la region
pX1,pY1,pX2,pY2	Long	Position de l'ellipse
[pType] = 0	Long	Type de positionnement Si pType = 0 : On passe un rectangle en paramètre, l'ellipse remplit ce rectangle pX1,Y1 : Point Haut-Gauche du rectangle pX2,Y2 : Point Bas-Droite du rectangle Si pType = 1 : on passe le centre et les rayons de l'ellipse en paramètre pX1,pY1 : Centre de l'ellipse pX2 : Rayon horizontal pY2 : Rayon vertical

IX-B - CreateRegionPolygon

Ajoute une région polygonale.

Pensez à fermer le polygone pour créer une région.

Paramètres :

Paramètre	Type	Explication
pRegion	String	Identifiant de la region
pPoints	Variante	Tableau de points formant le polygone Array(X1,Y1,X2,Y2,X3,Y3,...)

IX-C - CreateRegionRect

Ajoute une région rectangulaire.

Paramètres :

Paramètre	Type	Explication
pRegion	String	Identifiant de la region
pX1,pY1	Long	Point Haut-Gauche du rectangle
pX2,pY2	Long	Point Bas-Droite du rectangle

IX-D - DeleteRegion

Supprime une région.

Paramètres :

Paramètre	Type	Explication
pRegion	String	Identifiant de la region à supprimer

IX-E - DeleteAllRegions

Supprime toutes les régions.

IX-F - FrameRegion

Dessine un trait qui encadre la région.

Paramètres :

Paramètre	Type	Explication
pRegion	String	Identifiant de la region
[pPenColor] = vbBlack	Long	Couleur du trait
[pPenWidth] = 1	Long	Épaisseur du trait
[pDash] = 0	Boolean	Pointillés si Vrai
[pAlpha] = 255	Integer	Transparence du trait (0 --> 255)
[pInside] = Faux	Boolean	Encadre à l'intérieur si Vrai

IX-G - FillRegion

Remplit une region d'une couleur.

Paramètres :

Paramètre	Type	Explication
pRegion	String	Identifiant de la region
[pColor] = vbBlack	Long	Couleur de remplissage
[pColorGradient] = -1	Long	Deuxième couleur pour dégradé
[pGradientVert] = Faux	Boolean	Mettre à Vrai pour un dégradé vertical
[pAlpha] = 255	Integer	Transparence du remplissage (0 --> 255)

IX-H - HatchRegion

Hachure une région.

Paramètres :

Paramètre	Type	Explication
pRegion	String	Identifiant de la region
[pColor] = vbBlack	Long	Couleur de remplissage
[pBackColor] = -1	Long	Deuxième couleur pour le fond
[pStyle] = 0	Long	Type de hachure (énumération EGdipHatchStyle)
[pAlpha] = 255	Integer	Transparence des hachures(0 --> 255)

IX-I - GetRegionXY

Identifie la region située sur un point.

Paramètres :

Paramètre	Type	Explication
pX	Long	Position X du point
pY	Long	Position Y du point
[pInclude]	Variant	Tableau de régions à inclure
[pExclude]	Variant	Tableau de régions à exclure

La fonction renvoie le nom de la région située sur le point précisé.

Exemple :

Recherche l'identifiant de la région sous le curseur de la souris

```
Private Sub Image0_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Dim lRegion As String ' Region survolée
    ' On convertit les coordonnées vers des coordonnées images en pixels avec CtrlToImgX et CtrlToImgY
    lRegion = ClGdip.GetRegionXY(ClGdip.CtrlToImgX(X, Me.Image0), ClGdip.CtrlToImgY(Y, Me.Image0))
End Sub
```

IX-J - PointInRegion

Teste si un point est dans une region.

Paramètres :

Paramètre	Type	Explication
pX	Long	Position X du point
pY	Long	Position Y du point
pRegion	String	Nom de la région à tester

La fonction renvoie Vrai si le point est dans la région précisée en paramètre.

IX-K - GetRegionRect

Coordonnées rectangulaires d'une région.

Paramètres :

Paramètre	Type	Explication
pRegion	String	Identifiant de la région.
pX1, pY1, pX2, pY2	Long	Coordonnées du rectangle contenant la région.

IX-L - RegionExists

Teste si une région existe.

Paramètres :

Paramètre	Type	Explication
pRegion	String	Nom de la région à tester

La fonction renvoie Vrai si la région existe.

IX-M - RegionAddRegion

Ajoute une région à une autre région.

Paramètres :

Paramètre	Type	Explication
pRegion	String	Nom de la région cible
pRegionToAdd	String	Nom de la région à ajouter

La fonction renvoie Vrai si l'opération est réussie.

IX-N - RegionCombine

Combine deux régions.

Paramètres :

Paramètre	Type	Explication
pRegion1	String	Nom de la première région Cette région reçoit la région résultante de la combinaison Attention : La fonction FrameRegion ne fonctionnera plus pour cette région.
pRegion2	String	Nom de la deuxième région
pCombineMode	Long	Mode de combinaison Voir l'énumération EGdipCombineMode.

IX-O - CreateRegionFromColor

Ajoute une région définie par la couleur des points.

Attention : il n'est pas possible d'encadrer cette région avec la fonction FrameRegion.

Paramètres :

Paramètre	Type	Explication
pRegion	String	Nom de la région à créer
[pColor]=-1	Long	Couleur déterminant la région
[pAlpha]=-1	Long	Transparence déterminant la région
pIncludeColor	Boolean	- Vrai : La région contient les points de couleur pColor et de transparence pAlpha

		- Faux : La région contient les points de couleur différente de pColor et de transparence différente de pAlpha
[pPointX]=-1, [pPointY]=-1	Long	Si renseigné, seuls les points adjacents sont traités (à la manière du pot de peinture de paint)

IX-P - ScaleRegion

Redimensionne une région.

Paramètres :

Paramètre	Type	Explication
pRegion	String	Nom de la région
pScaleX, pScaleY	Single	Facteurs de redimensionnement

IX-Q - TranslateRegion

Déplace une région.

Paramètres :

Paramètre	Type	Explication
pRegion	String	Nom de la région
pdX, pdY	Single	Nombre de pixels de déplacement

IX-R - RotateRegion

Applique une rotation à une région.

Paramètres :

Paramètre	Type	Explication
pRegion	String	Nom de la région
pAngle	Single	Angle en degré
pCenter	Booléen	Si vrai, le rectangle contenant la région est recentré après rotation

IX-S - RegionsIntersect

Teste si deux régions ont une intersection.

La fonction renvoie Vrai si les deux régions ont une intersection.

Paramètres :

Paramètre	Type	Explication
pRegion1	String	Nom de la première région
pRegion2	String	Nom de la deuxième région Si ce paramètre est laissé vide, il sera renseigné avec la première région trouvée qui a une intersection avec pRegion1

IX-T - RegionClone

Clone une région.

Paramètres :

Paramètre	Type	Explication
pRegion	String	Nom de la région à cloner
pRegionClone	String	Nom de la région clone

IX-U - SetFormRegion

Limite l'affichage de formulaire a une région.

Paramètres :

Paramètre	Type	Explication
pForm	Formulaire	Le formulaire à limiter
pRegion	String	Nom de la région
[pCtrlRef]=Nothing	Image	Contrôle image de référence A utiliser si la région a été créée par rapport à l'image
pInverseRegion	Boolean	Si Vrai, la région est inversée
[pCorrectionLeft]=0	Single	Correction de position gauche de la région en pixel
[pCorrectionTop]=0	Single	Correction de position haute de la région en pixel
[pCorrectionWidth]=0	Single	Correction de largeur de la région en pixel
[pCorrectionHeight]=0	Single	Correction de hauteur de la région en pixel

IX-V - ResetFormRegion

Rétablit l'affichage du formulaire complet.

Paramètres :

Paramètre	Type	Explication
pForm	Formulaire	Le formulaire à rétablir

X - Sauvegarde d'images en mémoire

X-A - KeepImage

Sauvegarde l'image en mémoire

Paramètres :

Paramètre	Type	Explication
[PImage]="Default"	String	Nom de la sauvegarde

L'image est sauvegardée en mémoire, prête à être rétablie avec la fonction **ResetImage**.

X-B - ResetImage

Rétablit l'image de la mémoire

Paramètres :

Paramètre	Type	Explication
[PImage]="Default"	String	Nom de la sauvegarde

X-C - ImageExists

Test si une image sauvegardée existe en mémoire.

Paramètres :

Paramètre	Type	Explication
[PImage]="Default"	String	Nom de la sauvegarde à tester

X-D - DeletImage

Supprime une image en mémoire.

Paramètres :

Paramètre	Type	Explication
[PImage]="Default"	String	Nom de la sauvegarde à supprimer

X-E - DeleteAllImages

Supprime toutes les images en mémoire.

Paramètres :

X-F - GetImageBitmap

Renvoie le bitmap gdiplus d'une image.

Paramètres :

Paramètre	Type	Explication
[PImage]="Default"	String	Nom de la sauvegarde à supprimer
[pClone]= Faux	Boolean	Si Vrai, clone le bitmap.

XI - Gestion des données Exif

XI-A - GetExifData

Renvoie la valeur du tag EXIF.

Paramètres :

Paramètre	Type	Explication
pTag	Long	Identifiant du tag Exif à lire Voir l'énumération EGdipTagName

Renvoie la valeur du tag dans un Variant.

Renvoie Null si le tag n'existe pas.

XI-B - SetExifData

Ecrit certains tags Exif.

Paramètres :

Paramètre	Type	Explication
pTag	Long	Identifiant du tag Exif à lire Voir l'énumération EGdipTagName
pValue	Variant	Valeur du tag à écrire

Si la valeur passée en paramètre est **Null**, le tag est supprimé de l'image.

XI-C - GenerateThumbnail

Génère une miniature EXIF intégrée à l'image.

Paramètres :

Paramètre	Type	Explication
[pSize] = 200	Long	Taille maxi de la miniature.
[pQuality] = -1	Long	Qualité de la miniature (0 à 100)

XI-D - ImportExifData

Importe les données Exif d'un fichier dans l'image en cours.

Paramètres :

Paramètre	Type	Explication
pFile	String	Nom du fichier.
[pExcludeTags]	Tableau de Long	Tableau de tags Exif à exclure

XII - Gestion des Gif animés

XII-A - GifSetFrame

Change l'image courante d'un gif animé.

Paramètres :

Paramètre	Type	Explication
pFrame	Long	Numéro de l'image

XII-B - GifGetFrameCount

Nombre d'images d'un gif animé.

XII-C - GifGetFrameDelay

Delais d'affichage des images d'un gif animé.

Paramètres :

Renvoie un tableau contenant autant de délais que d'images dans le gif.

XIII - Gestion des textures

Utilisez ensuite les propriétés **FillTexture** et **PenTexture** pour choisir la texture à utiliser.

XIII-A - TextureAddFromFile

Ajoute une texture à partir d'un fichier.

Paramètres :

Paramètre	Type	Explication
pName	String	Identifiant de la texture
pFile	String	Chemin du fichier de la texture

XIII-B - TextureAddFromImage

Ajoute une texture à partir d'une image.

Paramètres :

Paramètre	Type	Explication
pName	String	Identifiant de la texture
plmgName	String	Image de la liste d'images contenant la texture

XIII-C - TextureAddFromControl

Ajoute une texture à partir de l'image intégrée a un contrôle.

Paramètres :

Paramètre	Type	Explication
pName	String	Identifiant de la texture
pCtrl	Image	Contrôle image contenant la texture

XIII-D - TextureDel

Supprime une texture.

Paramètres :

Paramètre	Type	Explication
pName	String	Identifiant de la texture

XIII-E - TextureTranslate

Déplace une texture.

Paramètres :

Paramètre	Type	Explication
pName	String	Identifiant de la texture
pDx, pDy	Single	Déplacements en pixels

XIII-F - TextureWidth

Largeur une texture.

Renvoie la largeur en pixel.

Paramètres :

Paramètre	Type	Explication
pName	String	Identifiant de la texture

XIII-G - TextureHeight

Hauteur une texture.

Renvoie la hauteur en pixel.

Paramètres :

Paramètre	Type	Explication
pName	String	Identifiant de la texture

XIV - Gestion des polices de caractères

Utilisez ensuite les propriétés **FillTexture** et **PenTexture** pour choisir la texture à utiliser.

XIV-A - FontPrivateAddFromFile

Ajoute une police de caractères privée à partir d'un fichier.

Cette police de caractères ne sera utilisable que dans l'instance de gdiplus où elle a été chargée.

Paramètres :

Paramètre	Type	Explication
pFile	String	Chemin du fichier de la texture

XIV-B - FontPrivateAddFromByte

Ajoute une police de caractères privée à partir d'une image.

Cette police de caractères ne sera utilisable que dans l'instance de gdiplus où elle a été chargée.

Paramètres :

Paramètre	Type	Explication
pByte	tableau de Byte	tableau de Byte contenu les données de la police de caractères

XIV-C - FontsPrivateDelete

Supprime toutes les polices de caractères privées.

Pas de paramètre

XIV-D - FontsEnumerate

Enumère les polices de caractères.

Paramètres :

Paramètre	Type	Explication
pFonts	Collection	Collection qui contiendra la liste des polices de caractères.

XV - Les fonctions de conversion

XV-A - ImgToCtrlX et ImgToCtrlY

Converti depuis les coordonnées image vers les coordonnées contrôle

On donne des pixels en entrée, on reçoit des points (Excel) ou Twips (Access) en sortie.

Paramètres :

Paramètre	Type	Explication
pX ou pY	Long	Abcisse ou ordonnées à convertir
pImgCtrl	Image	Contrôle Image à utiliser pour la conversion.
[pDecalage] = True	Boolean	Calcul avec décalage (Position de l'image dans le contrôle) Définir à Vrai pour convertir une position, à Faux pour convertir une taille.

XV-B - CtrlToImgX et CtrlToImgY

Converti depuis les coordonnées contrôle vers les coordonnées image

On donne des points (Excel) ou Twips (Access) en entrée, on reçoit des pixels en sortie.

Paramètres :

Paramètre	Type	Explication
pX ou pY	Long	Abcisse ou ordonnées à convertir
pImgCtrl	Image	Contrôle Image à utiliser pour la conversion.
[pDecalage] = True	Boolean	Calcul avec décalage (Position de l'image dans le contrôle) Définir à Vrai pour convertir une position, à Faux pour convertir une taille.

XV-C - PointsToPixelsX et PointsToPixelsY

Converti les Points (Excel) ou Twips (Access) en Pixels

Paramètres :

Paramètre	Type	Explication
pPointsX pPointsY	Long	Valeur à convertir

La fonction retourne la valeur convertie en **Pixels**

XV-D - PixelToPointsX et PixelToPointsY

Converti les Pixels en Points (Excel) ou en Twips (Access)

Paramètres :

Paramètre	Type	Explication
pPixelsX pPixelsY	Long	Valeur à convertir

La fonction retourne la valeur convertie en Points (Excel) ou en Twips (Access)

XVI - Les autres fonctions utiles

XVI-A - GetPixel

Renvoie la couleur d'un point

Paramètres :

Paramètre	Type	Explication
pX,pY	Long	Coordonnées du point

la fonction renvoie la couleur du point dans un Long.

Exemple :

Recherche la couleur du pixel sous le curseur de la souris

```
Private Sub Image0_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single) <br/>
    Dim lColor As Long ' Couleur sous la souris
    ' On convertit les coordonnées vers des coordonnées images en pixels avec CtrlToImgX et CtrlToImgY
    lColor = ClGdip.GetPixel(ClGdip.CtrlToImgX(X, Me.Image0), ClGdip.CtrlToImgY(Y, Me.Image0))
End Sub
```

XVI-B - LongToRGB

Conversion code couleur Long vers RGB

Paramètres :

Paramètre	Type	Explication
pLong	Long	Valeur de la couleur
pRed	Long	Composante rouge de la couleur
pGreen	Long	Composante verte de la couleur
pBlue	Long	Composante bleue de la couleur

XVI-C - SetXPTheme

Active ou désactive le thème XP pour les contrôles.

Permet de réduire le scintillement de l'image (pour WinXP/Acc2003)

La fonction s'applique sur toute l'application, mais seulement sur les contrôles (les barres de menu, barres de titre, ... conservent le thème XP).

Voir la fonction **SetDoubleBufferXP** pour réduire les scintillements sans désactiver le thème XP.

Utilisez cette fonction **même si vous avez déjà désactivé le thème XP dans les options de la base de données.**

Paramètres :

Paramètre	Type	Explication
[pActive]=True	Boolean	Vrai pour activer. Faux pour désactiver.

Exemple :

Réduire les scintillements si la version d'Access est 2003

```
' Désactive le thème XP pour réduire les scintillements si la version d'Access est 2003
If SysCmd(acSysCmdAccessVer) = "11.0" Then ClGdip.SetXPTheme False
```

XVI-D - SetDoubleBufferXP

Permet de réduire le scintillement de l'image (pour WinXP/Acc2003)

La fonction de double buffer est **gourmande en ressource**.

Envisagez l'utilisation de la fonction **SetXPTheme** pour **désactiver le thème XP**.

Paramètres :

Paramètre	Type	Explication
pForm	Formulaire	Formulaire dont on veut réduire le scintillement Privilégiez un formulaire de type indépendant (Fen. Indépendante = Oui). Si la fonction est appliquée sur un formulaire avec la propriété Fen. Indépendante = Non , alors le double buffer s'applique sur toute l'application et peut faire scintiller d'autres objets.
[pActive]=True	Boolean	Vrai pour activer. Faux pour désactiver.

Exemple :

Réduire les scintillements si la version d'Access est 2003

```
' Activer le double buffer pour réduire les scintillements si la version d'Access est 2003
If SysCmd(acSysCmdAccessVer) = "11.0" Then ClGdip.SetDoubleBufferXP Me
```

XVI-E - DragForm

Permet le déplacement du formulaire

Cette fonction permet de déplacer le formulaire même s'il n'a pas de barre de titre.

Exemple :

Paramètre	Type	Explication
pForm	Formulaire	Formulaire à déplacer

Comment déplacer le formulaire en cliquant sur une image

```
' Remarque : Ne fonctionne pas sur l'événement Click
Private Sub Image0_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    ClGdip.DragForm Me
End Sub
```

XVI-F - FontSizeToPixel

Conversion taille police en pixel

Convertit une taille de police de caractères en pixel pour la fonction **DrawText**.

Paramètres :

Paramètre	Type	Explication
pSize	Long	Taille de police
pCtrlRef	Control	Contrôle de référence

XVI-G - GetUserFormHandle

Identifiant d'un userform

Pour Excel.

Paramètres :

Paramètre	Type	Explication
pForm	Long	Objet formulaire
pClientArea	Boolean	Si Faux, la fonction renvoie l'identifiant du formulaire Si Vrai, la fonction renvoie l'identifiant de la sous-fenêtre sur laquelle on dessine

XVI-H - GetClientHandle

Identifiant de la sous-fenêtre sur laquelle on dessine (=section).

Pour Access.

Paramètres :

Paramètre	Type	Explication
plmgCtrl	Image	Objet image contenu dans la section de laquelle on veut l'identifiant.

XVII - Les propriétés

XVII-A - ImageWidth et ImageHeight

Taille de l'image en pixels

Type Long en Lecture seule.

C'est la taille de l'image en mémoire.

XVII-B - ImageFormat

Format de l'image

Type string en Lecture seule.

Formats possibles :

Undefined / MemoryBMP / BMP / EMF / WMF / JPEG / PNG / GIF / TIFF / EXIF / ICON.

Dépend du format du fichier ouvert avec **OpenFile**.

Si l'image est créée avec **CreateBitmap** ou modifiée après chargement (hors données Exif), le format de l'image est **MemoryBMP**.

XVII-C - RefControl

Contrôle de référence pour coordonnées passées en points/twips

Type Control en Lecture/Ecriture.

Si RefControl est défini, les fonctions acceptent des coordonnées passées en Twips (pour Access) ou en Points (Pour Excel).

Pour annuler et repasser en pixels, affectez **Nothing** à cette propriété.

Exemple :

Affichage d'un pixel rouge à l'emplacement de la souris

```
' Deux solutions identiques :
```

Affichage d'un pixel rouge à l'emplacement de la souris

```
'
' 1 - Sans utiliser RefControl
clGdip.DrawPixel clGdip.CtrlToImgX(X, Me.Image0), clGdip.CtrlToImgY(Y, Me.Image0), vbRed
'
' 2 - En utilisant RefControl
clGdip.RefControl = Me.Image0
clGdip.DrawPixel X, Y, vbRed
clGdip.RefControl = Nothing
```

XVII-D - PenAlignMode

Alignement du crayon

Type Long en Lecture/Ecriture.

Voir énumération **EGdipPenAlignMode**

S'applique aux fonctions DrawRectangle, DrawLine, DrawEllipse et DrawPolygon.

XVII-E - FastRepaintReplace

Mode remplace pour FastRepaint

Type Boolean en Lecture/Ecriture.

Si **FastRepaintReplace** est vrai, **FastRepaint** remplace les pixels.

Sinon ils sont dessinés par dessus l'image déjà affichée.

XVII-F - WrapX et WrapY

Décalage d'une image avec enroulement.

Type Long en Lecture/Ecriture.

S'applique aux fonctions **FastRepaint**, **DrawImage** et **DrawBitmap**.

XVII-G - DrawSmooth

Dessin avec lissage.

Type Booléen en Lecture/Ecriture.

S'applique aux fonctions de dessin de lignes, polygones et ellipse, ainsi qu'à l'encadrement de région (FrameRegion).

XVII-H - DrawWithoutTransparency

Dessin des images sans transparence (plus rapide si pas besoin de transparence).

Type Booléen en Lecture/Ecriture.

S'applique aux fonctions de dessin de lignes, polygones et ellipses.

XVII-I - FillTexture

Texture de remplissage.

Type String en Lecture/Ecriture.

S'applique aux fonctions de dessin de textes, rectangles, polygones et ellipses, ainsi qu'aux fonctions de remplissage (FillColor et FillRegion) .

XVII-J - PenTexture

Texture de crayon pour les lignes.

Type String en Lecture/Ecriture.

S'applique aux fonctions de dessin de rectangles, polygones et ellipses, ainsi qu'à l'encadrement de région (FrameRegion).

XVIII - Conclusion

La librairie gdiplus.dll offre de nombreuses possibilités graphiques que l'on peut exploiter dans Access. Merci à l'équipe Office de developpez.com pour ses relectures, commentaires et encouragements!

XIX - Les téléchargements

Télécharger la classe au format texte ([HTTP](#))

Ensuite insérez le contenu du fichier dans un module de classe.

Télécharger la base d'exemples au format ACCESS 2000 ([HTTP](#))

Télécharger la base d'exemples au format EXCEL 2000 ([HTTP](#))