

# **Fonctions de SAA.DLL**

Si vous utilisez Visual Basic (et que SAA.DLL est ajouté aux références), vous pouvez voir la liste des fonctions (la DLL contient 2 classes publiques) et des paramètres en appuyant sur F2 et en choisissant 'SAA'. Sinon voici la liste :

**Classe *.Application* :**

**About**  
**Annule**  
**APropos**  
**ConvASCIIVersImg**  
**ConvCoul**  
**ConvImgVersASCII**  
**GenerateurTexte**  
**Luminosite**  
**OuvrirSAA**  
Event **Progression**

## **Fonction About :**

Alias de la fonction APropos.

## **Fonction Annule :**

Annule l'opération en cours.

Utilisation : `Call object.Annule`

## **Fonction APropos :**

Affiche le copyright de la DLL.

Utilisation : `Call object.APropos`

## **Fonction ConvASCIIVersImg :**

Convertit un dessin ASCII (fichier texte) en une image (BMP). Renvoie True si la conversion s'est bien passée.

Utilisation : `VarBoolean = object.ConvASCIIVersImg(sCheminTXT As String, sCheminIMG As String, bRendu As Byte, dDeformeHauteur As Double, sLesCaracteres As String, [MontreProgression As Boolean = Vrai])`

Renvoi *True* si l'opération est réussi ou *False* sinon.

Détail des paramètres :

sCheminTXT	Chemin du fichier texte source.
sCheminIMG	Chemin du fichier image BMP destination.
bRendu	Rendu de l'image, valeurs possibles : 0 - Normal 1 - Inverse 2 - Rouge 3 - Vert 4 - Bleu 5 - Foncé 6 - Clair 7 - Space
dDeformeHauteur	Valeur entre 0 et 1 (pas déformée), 0.69 par défaut.
sLesCaracteres	Caractères à utiliser, exemple : MHI:.
MontreProgression rapidité.	Facultatif, mettre False ne montrera pas la progression mais augmentera la rapidité.

### **Fonction ConvCoul :**

Donne une couleur correspondant à la luminosité donné (en fonction du rendu choisi).

Utilisation : Couleur = *object*.ConvCoul(bLum As Byte, bRendu As Byte)

Détail des paramètres :

bLum	Luminosité
bRendu	Rendu , valeurs possibles : 0 - Normal 1 - Inverse 2 - Rouge 3 - Vert 4 - Bleu 5 - Foncé 6 - Clair 7 - Space

### **Fonction ConvImgVersASCII :**

Convertit une image (BMP, JPEG, GIF, ...) en dessin ASCII.

Utilisation : VarBoolean = *object*.Function ConvImgVersASCII(sCheImg As String, sCheTxt As String, iZoom As Double, dDeformeHauteur As Double, sLesCaracteres As String, [MontreProgression As Boolean = Vrai])

Renvoi *True* si l'opération est réussie ou *False* sinon.

Détail des paramètres :

sCheImg	Chemin du fichier image source.
sCheTxt	Chemin du fichier texte destination.
iZoom conseillé (50%).	Redimensionnement de l'image, 1 pour aucun redimensionnement, 0.5
dDeformeHauteur	Valeur entre 0 et 1 (pas déformée), 0.69 par défaut.
sLesCaracteres	Caractères à utiliser, exemple : MHI:.

*MontreProgression*      Facultatif, mettre False ne montrera pas la progression mais augmentera la rapidité.

### **Fonction GenerateurTexte :**

Génère un texte avec des caractères dessinées en ASCII Art à partir d'un texte de départ.

Utilisation : `VarString = object.GenerateurTexte(sTxt As String, bRendu As Byte)`

Détail des paramètres :

sTxt	Texte à générer
bRendu	Rendu , valeurs possibles :
	1 - Futuriste
	2 - Block
	3 - WireFrame

### **Fonction Luminosite :**

Donne la luminosité d'une couleur.

Utilisation : `ValeurByte = object.Luminosite(Coul As ColorConstants)`

Détail des paramètres :

Coul	Couleur
------	---------

### **Fonction OuvrirSAA :**

Démarre Supra ASCII Art s'il est installé et ouvre le fichier OuvrirUnFichier éventuellement.

Utilisation : `ValeurByte = object.OuvrirSAA([OuvrirUnFichier As String])`

Détail des paramètres :

<i>OuvrirUnFichier</i>	Facultatif, fichier texte à ouvrir au démarrage.
------------------------	--

### **Événement Progression :**

Est déclenché pendant une opération pour en indiquer sa progression.

Utilisation : `Progression(PourcentageAccompli As Single)`

Exemple de déclaration :

```
Private Sub object_Progression(PourcentageAccompli As Single)
'code
End Sub
```

Détail des paramètres :

PourcentageAccompli Indique la progression en %.

**Classe .Video :**

**AnnuleOperation**

**Avi2Bmp**

**Bmp2Avi**

Event **ProgressionAVI2BMP**

Event **ProgressionBMP2AVI**

### **Fonction AnnuleOperation :**

Annule l'opération en cours.

### **Fonction Avi2Bmp :**

Convertit un fichier vidéo AVI en une série d'images BMP.

Utilisation : `VarBoolean = object.Avi2Bmp(sFichierAVI As String, sRepPourBMP As String)`

Renvoi *True* si l'opération est réussie ou *False* sinon.

Détail des paramètres :

sFichierAVI	Chemin du fichier AVI source.
sRepPourBMP	Répertoire où seront stockées les BMP créées.

### **Fonction Bmp2Avi :**

Convertir une série d'images BMP en un fichier AVI.

Utilisation : `VarBoolean = object.Bmp2Avi(FormhWnd As Long, sCheminPremierBMP As String, sCheminDestAVI, [iFPS As Integer = 24])`

Renvoi *True* si l'opération est réussie ou *False* sinon.

Détail des paramètres :

FormhWnd	Handle du formulaire appelant la fonction.
sCheminPremierBMP	Chemin du premier fichier BMP de la série.
sCheminDestAVI	Chemin du fichier AVI qui sera créé
iFPS	Facultatif, Indique le nombre d'images par seconde du nouveau fichier AVI.

**Note :** Le handle est requis car au début de l'opération, une fenêtre apparaîtra pour demander à l'utilisateur le codec à utiliser.

### **Événement ProgressionAVI2BMP :**

Est déclenché pendant la conversion Avi vers Bmp pour en indiquer la progression.

Utilisation : `ProgressionAVI2BMP (PourcentageAccompli As Single)`

Exemple de déclaration :

```
Private Sub object_ProgressionAVI2BMP (PourcentageAccompli As Single)
'code
End Sub
```

Détail des paramètres :

`PoucentageAccompli` Indique la progression en %.

### **Événement ProgressionBMP2AVI :**

Est déclenché pendant la conversion Bmp vers Avi pour en indiquer la progression.

Utilisation : `ProgressionBMP2AVI (FichierEnCours As Long)`

Exemple de déclaration :

```
Private Sub object_ProgressionBMP2AVI (FichierEnCours As Long)
'code
End Sub
```

Détail des paramètres :

`FichierEnCours` Indique le numéro du fichier BMP qui est traité. Pour afficher un pourcentage, vous devrez donc récupérer le nombre total de BMP.