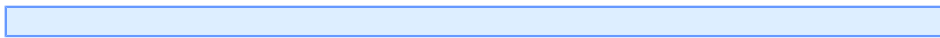


# Tutoriel 21 : Le contrôle "ANIBTN32.OCX"

par Gilbert Miralles ([gilmir.developpez.com](http://gilmir.developpez.com))

Date de publication : Lundi 18 mars 2003

Dernière mise à jour : Lundi 4 février 2008



- I - Visual Basic a révolutionné la programmation à deux niveaux.
  - II - Comment réaliser une animation sous VB sans écrire de code ? (ou si peu).
  - III - Télécharger
- Prochain tutoriel :

## I - Visual Basic a révolutionné la programmation à deux niveaux.

**La principale révolution** concerne la conception de l'interface utilisateur.

Son développement était complexe et représentait jusqu' à 75% du code.

L'approche graphique et "Visuelle" de Visual Basic permet désormais de concevoir une interface utilisateur simple et conviviale, en écrivant peu ou pas de code.

**La seconde révolution** est l'objet de ce Tutoriel.

Visual Basic est un langage de programmation modulaire.

Les modules qui composent son environnement sont présentés sous la forme de contrôles ou "VBX" pour les versions inférieures à VB4 et "OCX" qui sont apparues depuis le lancement de la version 32 bits de VB4.

Certains de ces contrôles livrés selon les versions disponibles sur le marché sont réutilisables à l'infini et complètent l'ensemble des outils fournis en standard avec Visual Basic.

Les puristes prétendent que Visual Basic n'est pas réellement un langage de programmation objet.

Pourtant, les principes majeurs de la POO (Programmation Orientée Objet), à savoir **l'encapsulation**, **l'héritage** et **le polymorphisme**, y sont présent.

Ces principes rendent possible une réutilisation du code supérieure aux langages conventionnels.

L' utilisation des contrôles signifie pour le programmeur, moins de code à écrire, donc simplifie au préalable l' écriture des applications développées.

J'avais dans l'esprit de vous initier aux développements de contrôles documentés avec l' utilisation d'un contrôle de type animation de boutons et comme l'application que je vais vous soumettre utilise des fichiers de sons, je suis dans l'obligation de faire d'une pierre deux coups et donc de vous présenter également le contrôle **MCI32.ocx**, mais vous n'allez pas vous en plaindre, n'est ce pas ?

Le contrôle **MCI32.ocx** est utilisé succinctement dans ce tutoriel et à la vue de ses nombreuses possibilités, il fera l'objet d'une étude plus approfondie ultérieurement.

Le contrôle Animation Boutons était livré en standard dans la version 3.0 de Visual Basic et se nommait "Anibutton.vbx", nous le retrouvons sous une plate forme 32 bits avec l'appellation usuelle de "**ANIBTN32.OCX**"

## II - Comment réaliser une animation sous VB sans écrire de code ? (ou si peu).

La qualité d'un bouton animé dépend de la qualité du graphisme employé pour la réalisation des images.

Réaliser quatre images qui représentent les étapes en développée des actions qui doivent être entreprises lorsque l'utilisateur provoque un événement, à savoir le bouton au repos en phase1, le bouton enfoncé en phase4 et deux phases intermédiaires ce qui va se traduire par une certaine souplesse dans le développée de l'animation.

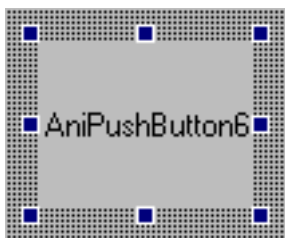
Si vous ne vous sentez pas en mesure de réaliser les images, vous pouvez les [télécharger ici](#).

Vous avez au préalable chargé dans votre boîte à outils le contrôle "OCX" ce qui a augmenté votre boîte à outils

d'une icône supplémentaire correspondant à cette figure graphique : 

### Comment fonctionne ce contrôle ?

Nous allons étudier le fonctionnement d' un objet tout en sachant qu'ils fonctionnent tous à l'identique.



Positionner un objet sur la feuille, il aura comme nom par défaut "AniPushButton1"

Nous allons modifier ses propriétés en fonction des événements que nous souhaitons provoquer.

**Name** : AniPushButton1

**Caption** : 1 (les boutons suivants auront la valeur de 2, puis 3, etc...)

**Cycle** : 0

**Height** : 675

**Width** : 495

**Picture** : Aller chercher l' image1 (bouton au repos)

**TextPosition** : 3- above (Positionner le texte sur le haut du bouton)

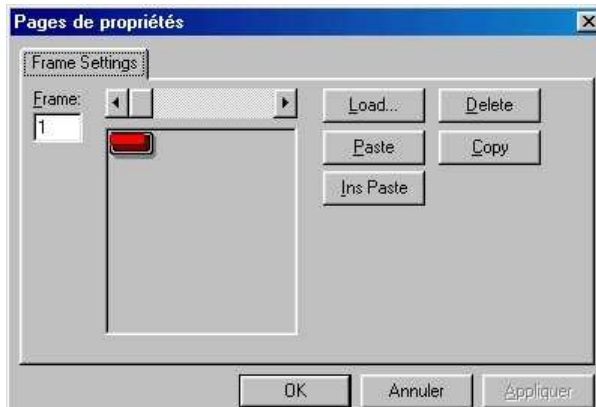
**Speed** : 80 (Vitesse de transition entre les images (80 ou 100))

Maintenant nous allons préparer l'animation du bouton en modifiant la propriété "Frame"que nous sélectionnons.

Un double clic sur cette propriété ouvre la boîte de dialogue Select Frame.

Cette dernière permet de construire la séquence d'animation.

Frame1 représente l'image1 que vous devez aller chercher par l'intermédiaire du bouton "Load"



L'image chargée s' affiche comme visualisée sur ce document.

En appuyant sur le bouton droite de l'ascenseur, vous incrémentez de un et passez à la frame2 et devez aller chercher l'image2 et ainsi de suite jusqu'à l'image4.

Vous continuez à charger les images en décrémentant depuis l'image 4 jusqu'à l'image 1 ce qui vous fait 8 images à charger pour la beauté de l' animation.

Lorsque toutes vos images sont chargées, vous refermez la fenêtre et l'animation doit fonctionner.

Vous pouvez télécharger le module exemple de l'application

Télécharger l'exo : [Exo1.zip](#)

Voici le prototype de terminé, vous pouvez faire des essais et appuyer sur les boutons pour voir se dérouler l'animation.



Cette image est un exemple de programme qui utilise des boutons animés pour simuler l'apparence et le fonctionnement d'une radio de voiture.

La ligne de boutons poussoirs est faite de 8 Bitmaps insérés dans le contrôle cadre.

Lorsque l'on appuie sur un bouton, celui-ci s'enfonce et sélectionne une station de radio, l'aiguille se déplace et la mélodie sélectionnée est diffusée.

La poussée des boutons est temporisée avec un décalage de 100 millisecondes d'intervalle, donnant l'illusion de mouvement de l'événement provoqué.

Les boutons **On/Off** et **Exit** sont créés avec 2 Bitmaps dans le cadre du contrôle.

Dans cet exemple il a été utilisé 5 contrôles **AniPushButton** pour les boutons poussoirs et 2 contrôles pour les boutons **On/Off** et **Exit**.

Trois images PictureBox pour la façade des stations de radio, un objet "Line" pour simuler l'aiguille et une image pour afficher la marque de l' autoradio.

Le boîtier de l'autoradio est simulé simplement avec des objets "Line", vous voyez que l'on peut se passer de "Frame" ou de contrôles similaires (Panel) qui utilisent de la mémoire et qui ne sont pas toujours indispensables, mais reconnaissons le, parfois tellement utiles.

### III - Télécharger

Le programme exemple **AnimButton**

Les **images** pour les boutons animés.

Prochain tutoriel :

 ***Protection de nos réalisations sous VB6***

