

Les Cases à Cocher et les Groupes d'Options dans Access

par Philippe JOCHMANS ([home page de Starec](#))

Date de publication : 03/12/2007

Dernière mise à jour : 03/12/2007

Ce tutorial a pour but de vous expliquer comment utiliser correctement les cases à cocher, les boutons d'option, les boutons bascule et les groupes d'options dans Access.

| | |
|--|----|
| I - Introduction..... | 3 |
| I-A - Généralité..... | 3 |
| I-B - Convention..... | 3 |
| I-C - Nommer les contrôles..... | 3 |
| II - Les Contrôles Indépendants..... | 4 |
| II-A - Les cases d'option et les cases à cocher..... | 4 |
| II-A-1 - Positionnement..... | 4 |
| II-A-2 - Valeurs..... | 4 |
| II-A-2-a - Affecter une valeur..... | 4 |
| II-A-2-b - Récupérer une valeur..... | 5 |
| II-B - Les boutons bascule..... | 5 |
| II-B-1 - Positionnement..... | 6 |
| II-B-2 - Valeurs..... | 6 |
| II-B-2-a - Affecter une valeur..... | 6 |
| II-B-2-b - Récupérer une valeur..... | 6 |
| II-B-3 - Personnaliser le bouton bascule..... | 6 |
| II-B-3-a - Légende (Caption)..... | 6 |
| III - Les Groupes d'Options..... | 6 |
| III-A - Créer un groupe d'option grâce à l'assistant..... | 7 |
| III-B - Renommer les éléments du groupe d'options..... | 9 |
| III-C - Connaître le bouton qui a été coché..... | 9 |
| III-D - Décider quel bouton sera activé..... | 10 |
| III-E - Rajouter une case d'option dans un groupe d'options déjà existant..... | 10 |
| IV - Conclusion..... | 10 |
| V - Remerciements..... | 11 |

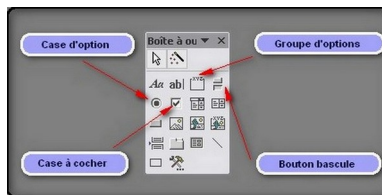
I - Introduction


I-A - Généralité

Le but de cet article est de vous expliquer comment utiliser correctement les groupes d'options, ainsi que les contrôles indépendants cases à cocher, cases d'option (également nommés boutons radio) et boutons bascule.

Nous n'allons pas étudier toutes les propriétés, et méthodes de ceux-ci, mais les plus importantes qui vont permettre de connaître leur comportement et leur état dans différentes situations.

Nous allons distinguer dans cet article les contrôles indépendants, et les contrôles dans les groupes d'options, en effet selon l'emplacement des contrôles, la récupération des données de ces contrôles sera différente.



 *Tous les tests ont été faits sous Access 2002.*

I-B - Convention

Par convention (vous pouvez le voir dans toutes les applications que vous utilisez et lorsque vous remplissez des questionnaires sur le Web), les contrôles indépendants ne sont pas utilisés n'importe comment.

- Les cases d'option : Ils sont utilisés lorsque vous n'avez qu'un choix à faire dans une liste.
- Les cases à cocher : Ils sont utilisés lorsque vous avez plusieurs choix possibles dans une liste.
- Les boutons bascule : Ils sont rarement utilisés dans les listes, leur avantage est plutôt visuel.

I-C - Nommer les contrôles

Lorsque vous disposez les contrôles sur un formulaire, Access donne à ceux-ci des noms par défaut, hors ceux-ci sont loin d'être parlant, il faut donc absolument les renommer.

Dans cet article nous allons utiliser les préfixes suivant pour les noms des contrôles :

- Les cases d'option : opt
- Les cases à cocher : chk
- Les boutons bascule : btn
- Les groupes d'options : opg

Je vous invite donc à nommer tout de suite vos contrôles, comme tous les éléments de votre base de données.

Je vous conseille la lecture de cet article :

Descriptif des conventions typographiques du code Visual Basic

II - Les Contrôles Indépendants

Nous allons évoquer dans cette partie les contrôles indépendants. Vous trouverez ceux-ci dans la barre d'outils lorsque votre formulaire est en mode création (voir image ci-dessus).

II-A - Les cases d'option et les cases à cocher

Nous allons regrouper ces deux contrôles ensemble, car ils ont exactement les mêmes propriétés, c'est uniquement l'aspect visuel qui change.


II-A-1 - Positionnement

Lorsque que vous positionner l'un de ces contrôles, celui-ci est accompagné d'une étiquette. Même si cette étiquette est purement à titre informatif, il vous appartient de bien définir son nom.

J'ai pris l'habitude de nommer les étiquettes en utilisant comme préfixe lbl, et en y ajoutant le nom du contrôle, ce qui permet au premier coup d'oeil de savoir que c'est l'étiquette du contrôle X.

Pour positionner correctement votre contrôle vous avez :

- **Haut (Top)** : qui positionne votre contrôle par rapport au bord haut de la section du formulaire contenant celui-ci.
- **Gauche (Left)** : qui positionne votre contrôle par rapport au bord gauche de la section du formulaire contenant celui-ci.

 *Faites attention aux unités, dans la fenêtre propriété lorsque votre formulaire est en mode création, l'unité est le cm. Tandis que lorsque vous passez par du code VB, l'unité est le twips.*

*Vous disposer également des propriétés **Hauteur (Height)** et **Largeur (Width)**, mais celles-ci ont peut d'intérêt, car vous ne pouvez redimensionner le cercle ou le carré de ces éléments, leur dimension est fixe.*

II-A-2 - Valeurs

II-A-2-a - Affecter une valeur

Lorsque vous avez positionné votre contrôle sur votre formulaire et que vous passez celui-ci en mode formulaire, vous pouvez constater que celui-ci est grisé.

En effet, ces deux contrôles admettent trois valeurs :

- **Coché** : La case est cochée.
- **Décochée** : La case est décochée.
- **Indéterminée** : La case est grisée.

Vous avez deux manières pour déterminer celle-ci.

Par la fenêtre propriété :

Il suffit de mettre dans la propriété **Valeur** par défaut **Vrai** si vous voulez que le contrôle soit coché ou **Faux** si vous voulez que celui-ci soit décoché, si vous voulez que celui-ci soit grisé, supprimez la valeur de cette propriété.

Par le code VBA :

Il vous faut passer par la propriété **Value**, ainsi le code suivant va rendre votre contrôle coché.

```
Me.chkAvecImpression.Value = True
```

Pour que le contrôle soit décoché, vous utilisez **False**, et pour que celui-ci soit grisé vous mettez **Null**.

II-A-2-b - Récupérer une valeur


Maintenant que nous avons affecté une valeur à notre contrôle, nous allons voir comment la récupérer. Pour cela nous allons utiliser la propriété **Value** de celui-ci.

```
Public Function TextChk(objChek As Control) As String
    ' ===== cette fonction va permettre de récupérer la valeur d'un checkbox =====
    Select Case objChek.Value
        Case -1
            TextChk = "Vrai, la case est cochée"
        Case 0
            TextChk = "False, la case est décochée"
        Case Else
            TextChk = "Null, la case est grisée"
    End Select
End Function
```

Et par un bouton de commande sur le formulaire.

```
Private Sub cmdTestChek_Click()
    ' ===== récupération de la valeur d'une texte box =====
    MsgBox Nz(TextChk(Me.chkAvecImpression, "Null"))
End Sub
```

Cela vous permet de tester n'importe qu'elle case d'option ou bouton à cocher.

 *Dans le code ci-dessus, j'ai utilisé la fonction NZ, en effet une case à cocher peut avoir une valeur Null. Hors lorsque vous cliquez sur une case à cocher, vous vous apercevez qu'elle est soit cochée, soit décochée. En fait cela est du à une propriété qui est Triple état (TripleState), cette valeur est par défaut à Non, si vous la passez à Oui, la case à cocher peut à nouveau avoir une valeur nulle. Donc contrairement à ce que l'on pense, une case à cocher indépendante peut avoir 3 valeurs et non 2.*

II-B - Les boutons bascule

Nous allons aborder les boutons bascule en tant qu'objet indépendant, alors que son utilisation est plus intéressante graphiquement dans les groupes d'options. Mais il permet aussi de faire des choses très intéressantes.

Lorsque vous utilisez le bouton de commande classique celui-ci une fois cliquer revient à sa position, avec le bouton bascule nous allons pouvoir garder la position appuyée.

II-B-1 - Positionnement

Le positionnement des boutons bascule est similaire à celui des cases à cocher (d'ailleurs c'est la même chose pour tous les contrôles, sauf que le bouton bascule n'est pas associé à une étiquette).

Les propriétés hauteur et largeur ont par contre tout leur intérêt, car comme le bouton de commande classique, sa taille peut être définie dans la fenêtre propriété et par le code.

II-B-2 - Valeurs

II-B-2-a - Affecter une valeur

Le comportement est similaire aux cases à cocher, je vous renvoie sur ce chapitre pour plus d'informations.

II-B-2-b - Récupérer une valeur

Le comportement est similaire aux cases à cocher, je vous renvoie sur ce chapitre pour plus d'informations.

II-B-3 - Personnaliser le bouton bascule

Le bouton bascule peut-être personnalisé comme le bouton de commande.

II-B-3-a - Légende (Caption)

Cette propriété va vous permettre de déterminer le texte qui se trouve dans ce bouton (vous pouvez bien sûr modifier les attributs de celui-ci : police, taille, etc.).

Ce qui est intéressant, c'est que l'on peut changer le texte en fonction des trois états possibles de ce bouton à bascule, alors que ce n'est pas possible avec un bouton de commande (on peut changer le texte, mais cela implique d'avoir une variable supplémentaire pour stocker la position du bouton de commande).

Mettez la propriété Tripe état à Oui, ainsi le code suivant vous donne trois légendes possibles pour le bouton.

```
Private Sub btnBascule_Click()  
    ' ===== changement de la légende du bouton à bascule en fonction de son état =====  
    Select Case Me.btnBascule.Value  
        Case -1  
            ' le bouton est enfoncée  
            Me.btnBascule.Caption = "Le bouton est enfoncé"  
        Case 0  
            ' le bouton est relâché  
            Me.btnBascule.Caption = "Le bouton est relâché"  
        Case Else  
            ' la valeur du bouton est null  
            Me.btnBascule.Caption = "Le bouton est neutre"  
    End Select  
End Sub
```

III - Les Groupes d'Options

Nous avons abordé dans la première partie les contrôles indépendants, hors comme le nom l'indique, il n'y a aucune liaison entre eux.

Pour que ceux-ci soient liés, il faut les positionner dans un groupe d'options, ce qui permet de décocher les contrôles quand on en coche un autre.

III-A - Créer un groupe d'option grâce à l'assistant

L'assistant pour créer les groupes d'options est excellent, je vous conseille donc de l'utiliser. Il vous permet de réaliser rapidement un groupe d'options.

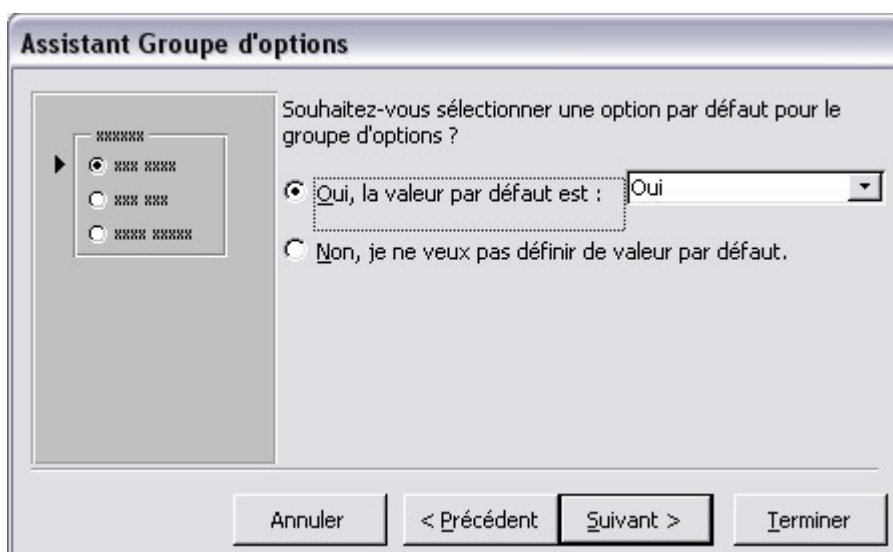
Pour pouvoir l'utiliser, il faut que l'icône de l'assistant soit enfoncé, ensuite vous n'avez plus qu'à cliquer sur l'icône groupe d'options et le positionner sur votre formulaire pour lancer l'assistant et effectuer les étapes qui vous sont proposés.

Nous allons créer grâce à l'assistant un groupe d'options avec des cases d'option OUI et NON.



Saisissez Oui dans le premier champ, appuyez sur la touche Tab, puis saisissez Non.


Ensuite cliquez sur suivant.



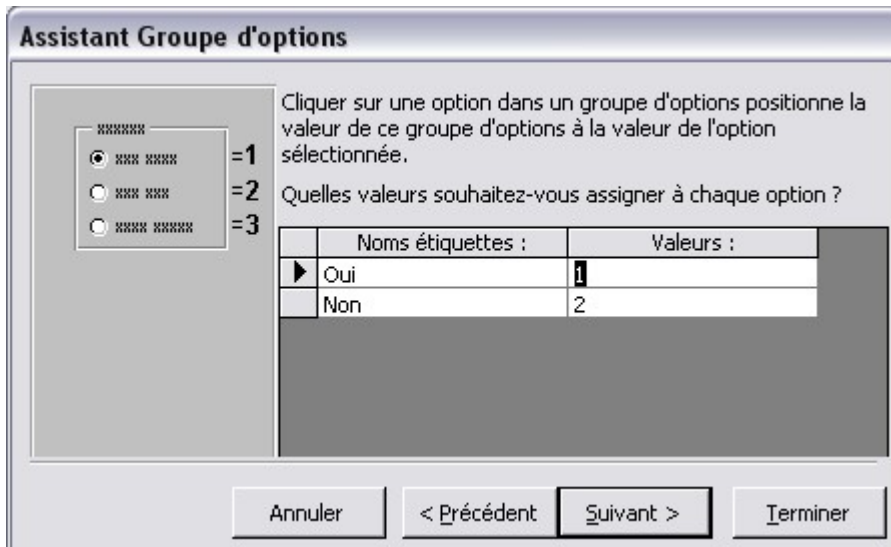
Maintenant vous allez déterminer quelle sera la valeur par défaut du groupe d'options.

Vous pouvez donc :

- Choisir une valeur par défaut dans une liste déroulante qui reprend les valeurs saisies précédemment.
- Ne pas choisir de valeur par défaut, ce qui implique que vos boutons d'options apparaîtront tous grisés.


 *Il est préférable de choisir une valeur par défaut.*

Puis cliquez sur suivant.

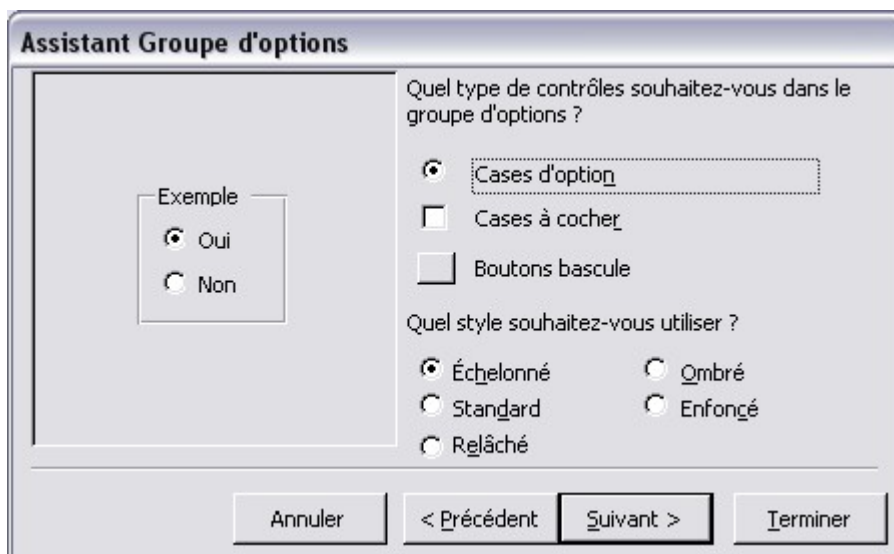


Cette fenêtre vous permet de déterminer la valeur que vous attribuer à chaque bouton d'option de votre groupe d'options.

Cette valeur nous permettra à l'avenir de savoir, en interrogeant le groupe d'option, quel bouton d'option a été coché.

 *Je vous conseille de laisser les valeurs prédéfinies par Access, cela vous permettra de vous y retrouver plus facilement par la suite en ayant une numérotation logique.*

Puis cliquez sur suivant.



Cette fenêtre va vous permettre de déterminer l'aspect de votre groupe d'options.

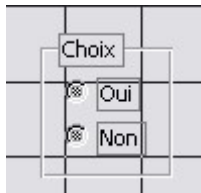
i Comme nous l'avons vu précédemment, la convention veut que si nous avons des boutons d'options qui doivent s'annuler quand nous cliquons sur un autre on utilise les cases d'option. Pour ne pas dérouter les utilisateurs, je vous conseille de prendre soit les cases d'option ou les boutons bascule, mais d'éviter les cases à cocher.

Puis cliquez sur suivant.



Maintenant, l'assistant vous demande de déterminer la légende qui vous voulez voir apparaître sur le cadre entourant votre groupe d'options. Par défaut, celui-ci vous propose le mot Cadre avec un chiffre, ici nous allons écrire Choix.

Puis cliquez sur Terminer.



Si vous passez en mode formulaire, vous avez dorénavant un groupe d'options parfaitement opérationnel, si vous cliquez sur Non (car nous avons pris Oui comme valeur par défaut), vous verrez le bouton Oui se décocher.

III-B - Renommer les éléments du groupe d'options

Comme il a déjà été dit, il est important de renommer vos contrôles. Le groupe d'options est composé de plusieurs éléments, nous allons maintenant les renommer.

- Le cadre : Le cadre est le conteneur du groupe d'options, nous allons le renommer : `opgChoix`.
- La légende du cadre : La légende du cadre (qui contient déjà Choix) est une étiquette, nous allons la renommer : `lblOpgChoix`.
- Les cases d'option : Il y a donc deux cases d'option, nous allons les renommer : `optOui` et `optNon`.
- Les légendes des cases d'options : Il y'a donc une légende par case, nous allons les renommer : `lblOptOui` et `lblOptNon`.

III-C - Connaître le bouton qui a été coché

Pour connaître la case qui a été cochée, il suffit de récupérer la propriété **value** du groupe d'options.

Le code suivant vous permet de récupérer la valeur de la case d'option lorsque vous cliquez sur l'une d'elle.

```
Private Sub opgChoix_Click()
    ' ===== récupération de la case cochée =====
    Select Case Me.opgChoix.Value
        Case 1
            MsgBox "Vous avez coché Oui"
        Case 2
            MsgBox "Vous avez coché Non"
    End Select
End Sub
```


L'utilisation la plus courante est de mettre ce code sur l'évènement Click d'un bouton de commande pour lancer un traitement en fonction du résultat.

III-D - Décider quel bouton sera activé

Il peut arriver que lors de votre traitement vous ayez besoin de changer le bouton qui sera activé, pour cela on utilise toujours la propriété **Value** du groupe d'options.

```
Me.opgChoix.Value = 2
```

Avec cette ligne de code nous allons cocher la case à cocher Non (ce qui décochera automatiquement la case Oui).

 Cette valeur peut-être redéfinie manuellement en changeant la propriété *Valeur par Défaut* (*DefaultValue*).


III-E - Rajouter une case d'option dans un groupe d'options déjà existant


Il est toujours possible d'ajouter une case d'option dans un groupe d'option sans avoir à le recréer.

Pour cela il vous suffit de sélectionner la case d'option dans la barre d'outil et de la positionner dans le groupe d'options. Vous verrez celui-ci se mettre en surbrillance.

Votre groupe d'options comporte maintenant 3 case à cocher Oui, Non et une autre que nous pourrions appeler Peut-être.

Ce dernier bouton prend pour valeur 3.

 Vous pouvez récupérer les valeurs des cases à cocher des groupes d'options en consultant la propriété **Valeur Contrôle** (*OptionValue*).

 La propriété **Triple état** (*TripleState*) n'existe pas pour les contrôles quand ceux-ci appartiennent à un groupe d'options.

IV - Conclusion

Comme vous l'avez vu, l'utilisation de ces contrôles ne se fait pas au hasard, voici donc un petit récapitulatif.

Vous voulez cocher ou décocher des contrôles indépendants les uns des autres :

Utilisez un contrôle case à cocher ou bouton bascule.

Vous voulez cocher ou décocher des contrôles en les liants les uns aux autres :

Utilisez les groupes d'options en sélectionnant des cases d'option ou des boutons bascule.

V - Remerciements

Je tiens à remercier **Dolphy35**, **Lou Pitchoun** et **RideKick** pour leurs relectures et corrections.

Et enfin toute l'équipe de **developpez.com** pour ce remarquable site.

