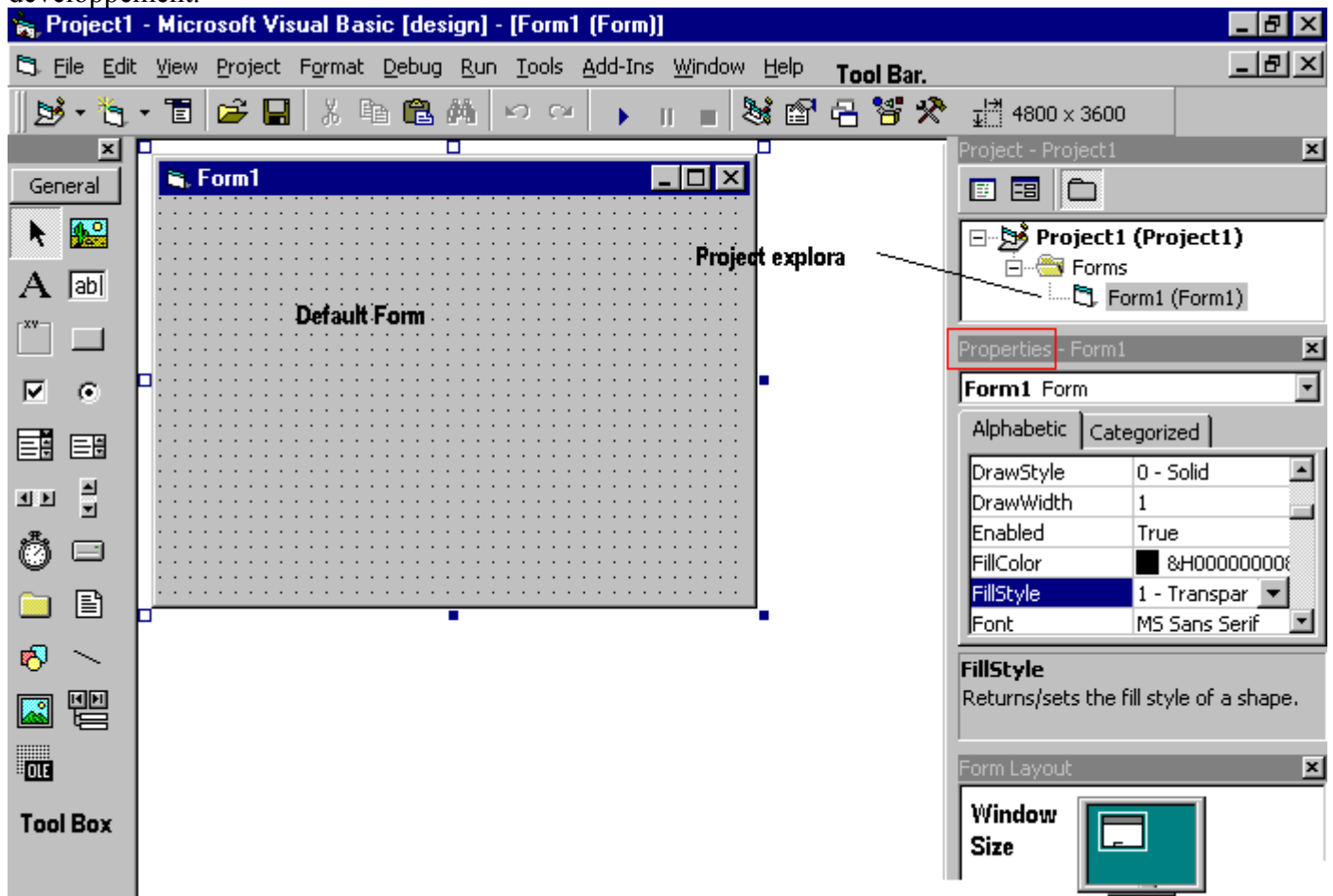


Leçon 1

L'environnement de développement

Apprendre les tenants et les aboutissants de l'environnement de développement avant d'apprendre Visual Basic est un peu comme l'apprentissage d'un test que vous devez savoir où toutes les fonctions appartiennent et quel est leur but. Nous allons d'abord commencer par l'étiquetage de l'environnement de développement.

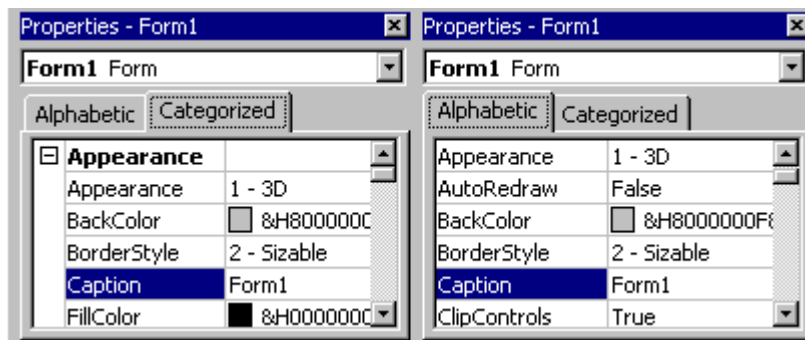


Le diagramme ci-dessus montre l'environnement de développement avec tous les points importants marqués. Beaucoup de fonctions de base de Visual travaux semblables à ceux de Microsoft Word par exemple, la **barre d'outils** et la boîte à outils est similaire à d'autres produits sur le marché qui travailler à l'extérieur d'un simple clic, puis faites glisser la largeur de l'objet requis. La **boîte à outils** contient le contrôle que vous avez placé sur la fenêtre du formulaire. Toutes les commandes qui apparaissent sur la **boîte à outils** des contrôles sur la photo ci-dessus n'est jamais à court de contrôles dès que vous placez l'un sur l'autre forme vous attend sur la **boîte à outils** prêts à être placés selon les besoins.

La fenêtre de l'explorateur de projet

La fenêtre de l'explorateur de projet vous donne une vue arborescente de tous les fichiers insérés dans l'application. Vous pouvez développer ces branches et de l'effondrement de l'opinion pour obtenir plus ou moins de détails (**explorateur de projet**). Le projet Explorer affiche la fenêtre des formes, des modules ou d'autres séparateurs qui sont pris en charge par Visual Basic comme class'es et Modules avancés. Si vous souhaitez sélectionner un formulaire sur son double-cliquez simplement sur la propre fenêtre de l'explorateur de projet pour un examen plus détaillé. Et il l'affiche où la **forme par défaut** est situé.

Fenêtre Propriétés



Certains programmeurs préfèrent la vue Categorized de la fenêtre de propriétés. Par défaut, la fenêtre de propriétés affiche ses propriétés par ordre alphabétique (à l'exception de la valeur du nom) lorsque vous cliquez sur le bouton classé les changements fenêtre à l'image de gauche.

La mise en page par défaut

Quand on commence à Visual Basic, nous sont fournis avec un projet VB projet. Un VB est une collection de modules et les fichiers suivants.

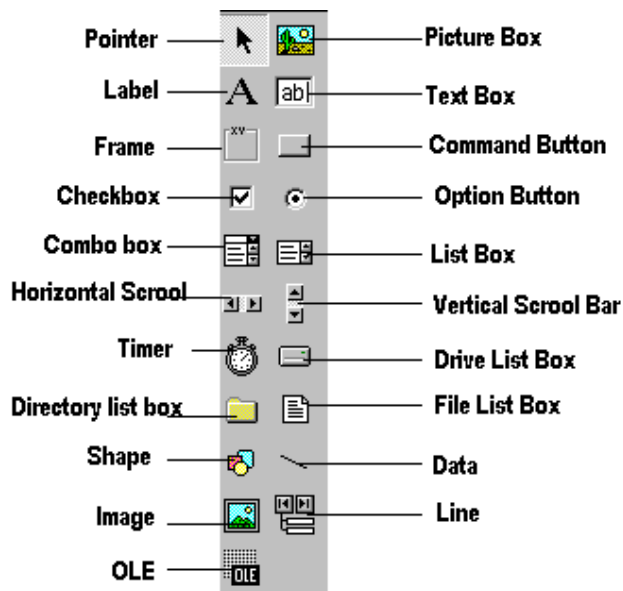
- Le **module global** (*qui contient la déclaration et les **procédures***)
- Le **module de formulaire** (*qui contient les éléments graphiques de l'application VB avec le **d'utilisation***)
- L' **module général** (*qui contient généralement des fins qui ne relèvent pas des instructions générales à tout graphique à l'écran*)
- L' **module de classe** (*qui contient les caractéristiques qui définissent une classe, y compris ses propriétés et méthodes*)
- L' **fichiers de ressources** (*qui vous permet de rassembler tous les textes et les images bitmap pour une application en un seul endroit*)


Au démarrage, Visual Basic affiche les fenêtres suivantes:

- La fenêtre de **formulaire vierge**
- La fenêtre **Projet**
- La fenêtre **Propriétés**

Il comprend également une **boîte à outils** qui se compose de tous les contrôles indispensables au développement d'une application VB. Les contrôles sont des outils tels que des boîtes, des boutons, des étiquettes et autres objets dessiner sur un formulaire pour obtenir ou **afficher d'entrée-sortie**. Ils ont également ajouter un attrait visuel.

Comprendre la boîte à outils.



Vous avez peut être remarqué que lorsque vous cliquez sur les différents contrôles de la **fenêtre Propriétés** change légèrement cela est dû à différents contrôles ayant des fonctions différentes. Par conséquent options supplémentaires sont nécessaires par exemple si vous aviez un tableau, vous souhaitez afficher une image. Mais si vous vouliez ouvrir une connexion internet vous devez remplir la machine distante et d'autres paramètres tels. Lorsque vous utilisez la commande (, Vous trouverez, un nouvel ensemble de propriétés sont les suivantes fournissent une description et une propriété.

Ouverture d'un projet Visual Basic existants.

Microsoft a inclus quelques freebies avec Visual Basic pour montrer ses capacités et ses fonctions. Démontage ou de modifier ces exemples de projets est une bonne façon de comprendre ce qui se passe à l'exécution. Ces fichiers peuvent être situés dans votre répertoire par défaut / **samples** /

Pour ouvrir ces projets choisissez "Ouvrir un projet» dans le menu «Fichier». Ensuite, double cliquez sur le dossier d'exemples d'ouvrir le répertoire, puis double-clic sur n'importe quel projet pour le charger.

Ouverture d'une nouvelle base de fichiers visuels et du code source Insertion.

En regardant les exemples il temps de faire votre propre application. Choisissez «Nouveau projet» dans le menu «Fichier». Utilisez le form1 vide pour concevoir une interface simple pour les agents immobiliers une base de données, ont des zones de texte pour les noms et autres détails. Insérer des contrôles et de faire un look professionnel. Zones de texte peut être utilisé pour y stocker le nom et les autres, assurez-vous de mettre une zone d'image dans une image de la maison.




Maintenant, insérez le code source suivant pour votre application.

Private Sub Form_Load ()

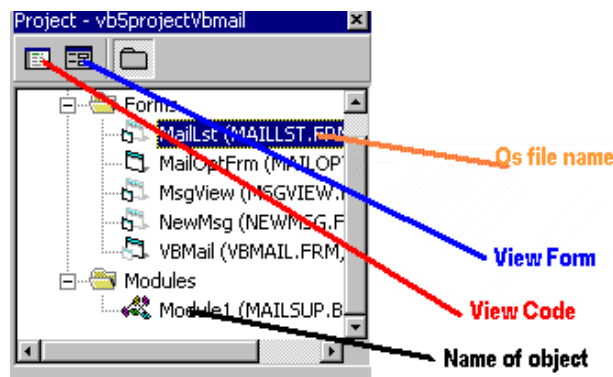
(": \ Program Files \ VB \ Graphics \ Icons \ Misc \ MISC42.ICO C") Picture1.Picture = LoadPicture

End Sub


Exécution et affichage du projet en détail.

Une fois la demande est chargé, il peut être exécuté en cliquant sur le  icône de la barre d'outils, d'appuyer sur pause  et de mettre fin à l'utilisation .

Une fois qu'un projet est chargé, au nom de la forme (s) qu'il contient est affichée dans la fenêtre du projet. Pour afficher un formulaire en mode Création, sélectionnez la forme requise en cliquant avec la souris pour mettre son nom, puis en cliquant sur le bouton du formulaire vue.



Dans cet exemple, le projet a été chargé et le maillist.frm a été sélectionné pour être affiché. Ce projet Mme exemple Mail useds 6 formes et 1 modules.

En mode Design, lorsque le formulaire est affiché, le code attaché à l'écran n'importe quel objet peut être inspecté par un double clic sur cet objet. Les captures d'écran ci-dessous montrent l'interface des échantillons Mail Mme (.../ exemple / Comtool / VBMail / MaiLLST.FRM) pour afficher le code pour cette forme sélectionnez  de l'élément de la fenêtre projet.

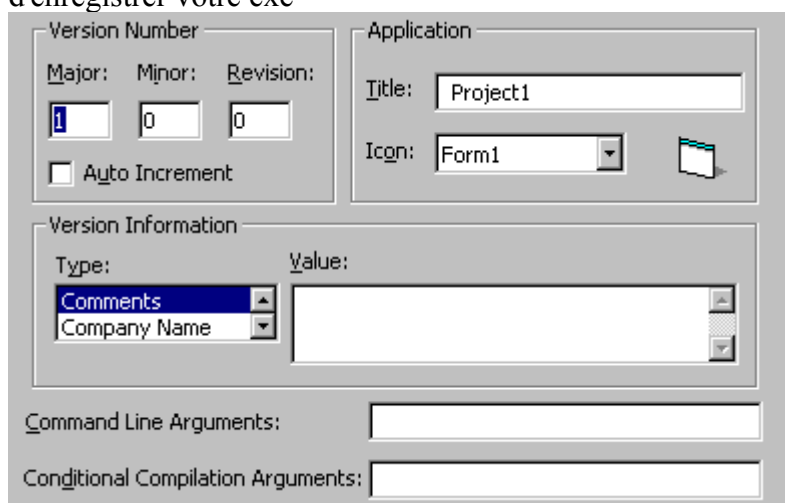
Private Sub SetupOptionForm (BasePic As Control)

```
BasePic.Top 0 =
BasePic.Left 0 =
BasePic.Visible = True
BasePic.enabled = True
OKBt.Top = BasePic.Height + 120
Me.width = BasePic.Width + 120
Me.Heigh OkBt.Top = + + 495 OkBt.Height
```

End Sub

Faire votre premier *. exe!?

Pour faire un exécutable d'un projet choisissez «Project.exe Makemake du menu" Fichier ". Puis cliquez une fois sur le Assurez Project.exe choisir un emplacement par défaut pour stocker votre fichier exécutable, vous pouvez également modifier des options avancées en cliquant sur les **options ..** tag avant d'enregistrer votre exe



L'image ci-dessus seront affichées en valeur le commentaire de certains commentaires entreprise nom de nom etc .. La balise Title représente la légende, vous verrez si vous appuyez sur Ctrl + Alt + Suppr et l'icône est l'icône qui sera disponible sur l'icône exécuter. Comme vous pouvez le voir c'est assez simple

à comprendre. Tous les commentaires, de données et le nom apparaît lorsque vous cliquez sur les propriétés et cliquez sur exe compilé (exécution).

L'enregistrement de votre base de projets visuels.

Enregistrez votre travail sur le disque. Utilisez l'Explorateur Windows ou toutes les fenêtres de bureau pour vérifier que tous les fichiers ont été sauvegardés. Il devrait y avoir une projet Visual Basic (. VBP) et le fichier formulaire distinct (. FRM) et le module (. BAS) des fichiers pour chaque forme et le module utilisé dans le projet actuel.

Propriétés des boutons de renvoi




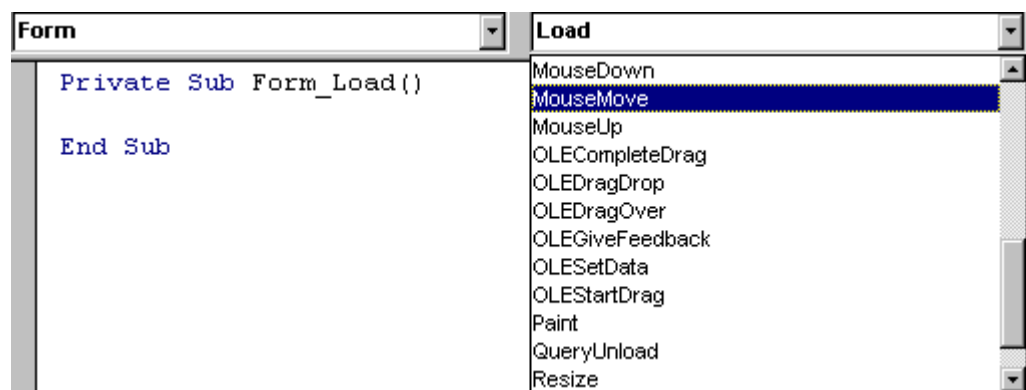
Bouton de commande et les propriétés des étiquettes

Propriété	Description
Nom	Le nom de l'objet afin que vous puissiez l'appeler à l'exécution
BackColor	Ceci spécifie la couleur de fond du bouton de commande. Cliquez <i>BackColor</i> de la palette de l'flèche vers le bas pour voir la liste des communes couleurs de configuration de Windows, vous devez modifier ce à la propriété de style de 0 - standard à 1 - graphiques
Annuler	Détermine si le bouton de commande reçoit un événement Click si l'utilisateur appuie échapper
Légende	Contient le texte qui apparaît sur le bouton de commande.
Par défaut	Détermine si le bouton de commande répond à une pression de touche entrer, même si un autre contrôle a le focus
Activer	Détermine si le bouton de commande est actif. Souvent, vous allez changer le permettre à la propriété à l'exécution avec le code de prévenir l'utilisateur en appuyant sur la touche
Font	Produit d'une police boîte de dialogue dans laquelle vous pouvez définir le nom de la légende de police, le style et la taille.
Hauteur	Positions de la hauteur de l'objet - peut être utilisé pour le bas
Gauche	Positions de la commande de gauche - peut être utilisé pour le droit
MousePointer	S'il est choisi pour une icône peut changer l'image du pointeur de la souris sur cet objet
Photo	Maintenez le nom d'un graphique image de l'icône de sorte qu'il apparaît comme une image au lieu d'un bouton pour que cette option fonctionne le tag graphique doit être mis à 1
Style	Cette détermine si le bouton de commande apparaît comme un standard de Windows boîte de dialogue ou une image graphique
Index de tabulation	Spécifie l'ordre du bouton de commande dans l'ordre de tabulation
Taquet de tabulation	Si l'objet peut être onglets pour (ce qui peut être utilisé sur les étiquettes qui n'ont pas d'autre fonction)
Texte de pointe	Si la souris est maintenu sur l'objet d'une brève description peuvent être affichés (par exemple tenez votre souris sur une des images ci-dessus pour voir ce qui se passe
Visible	Si vous voulez que l'utilisateur de voir le bouton / label sélectionnez vrai sage suffit d'appuyer sur d'autres faux
Largeur	Afficher la largeur de l'objet

Leçon 2

Comprendre les événements

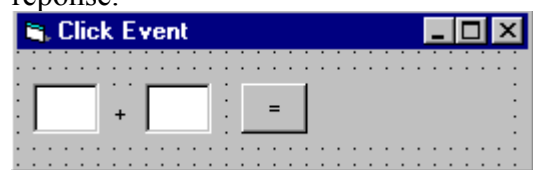
Le «look» d'une application Visual Basic est déterminé par ce que les contrôles sont utilisés, mais la «sensation» est déterminée par les événements. Un événement est quelque chose qui puisse arriver à un contrôle. Par exemple, un utilisateur peut cliquer sur un bouton, une zone de texte, ou redimensionner une forme. Comme expliqué dans la création d'une application Visual Basic, la rédaction d'un programme est constitué de trois événements: 1) sélectionner des contrôles appropriés, 2) définir les propriétés, et 3) d'écrire le code. Il est à l'écriture de code stade où il devient important de choisir des activités appropriées pour chaque contrôle. Pour ce faire, double-cliquez sur le contrôle de la manifestation sera l'occasion pour, ou cliquez sur le  icône dans la fenêtre du projet (à droite en général haut de l'écran). Une fenêtre de code devrait maintenant être affichée similaire à celui ci-dessous.



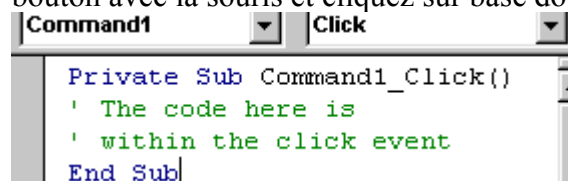
La boîte déroulante de gauche présente une liste de tous les contrôles utilisés par la forme actuelle, la forme elle-même, et une section spéciale intitulée Déclarations générales. La liste déroulante correspondante sur la droite affiche une liste de tous les événements applicables à la régulation du courant (tel que spécifié par la liste déroulante de gauche). Les événements affichés en caractères gras signifie que le code a déjà été écrit pour eux, les événements sont unbold utilisé. Pour démontrer que différentes manifestations peuvent jouer un rôle important dans la détermination du sens d'une demande, un programme de petit exemple sera écrite à additionner deux nombres et afficher la réponse. La première solution à ce problème utiliser l'événement clic d'un bouton de commande, tandis que la seconde sera l'événement de modification de deux zones de texte.

Cliquez sur l'événement

Avant les événements peuvent être codées, il est nécessaire de concevoir l'interface de contrôles appropriés. Comme le montre la capture d'écran ci-dessous utilisent: 2 zones de texte pour entrer les chiffres, un label pour le signe «+», un bouton de commande pour le signe «=», et un autre label pour la réponse.



Faire l'événement click est très simple il suffit de sélectionner le bouton avec la souris et cliquez sur base double visuelle va générer



Vous pouvez voir en haut à droite il ya une liste déroulante "clic" c'est ce qu'on appelle un gestionnaire d'événements.

Ecriture de votre propre même

Dans le premier exemple, l'utilisateur doit entrer deux nombres, puis cliquez sur le bouton total pour produire une réponse. Toutefois, le programme peut être modifié de sorte que la réponse sera calculé chaque fois que l'un des deux numéros sont changés sans nécessiter un bouton total.

Pour ce faire, retirez d'abord les égaux bouton de commande et de le remplacer par une étiquette avec la légende à '='. Maintenant, ouvrez une fenêtre de code et de copier dans le presse papier de Windows `lblAnswer ligne = Str $ (Val (txtNumber1.Text) + Val (txtNumber2.Text))`. Utilisation de la liste déroulante boîte de la main gauche, sélectionnez la première zone de texte, puis sélectionnez l'événement **Change** dans la liste déroulante de droite. Ensuite, collez le code de la presse-papiers dans le sous-programme vide. Sélectionnez la deuxième zone de texte et de faire la même chose. La même ligne est nécessaire à deux reprises parce que les deux événements de clic appartiennent à deux commandes séparées. Le code final devrait ressembler à:


```
Private Sub txtNumber1_Change ()
    Label2.Caption = Str $ (Val (Text1.Text) + Val (text.Text))
End Sub

Private Sub txtNumber2_Change ()
    Label2.Caption = Str $ (Val (Text1.Text) + Val (text2.Text))
End Sub
```

Exécutez le programme à nouveau, entrez les deux chiffres et observer ce qui se passe. Chaque fois qu'un changement chiffres, la réponse est recalculée.

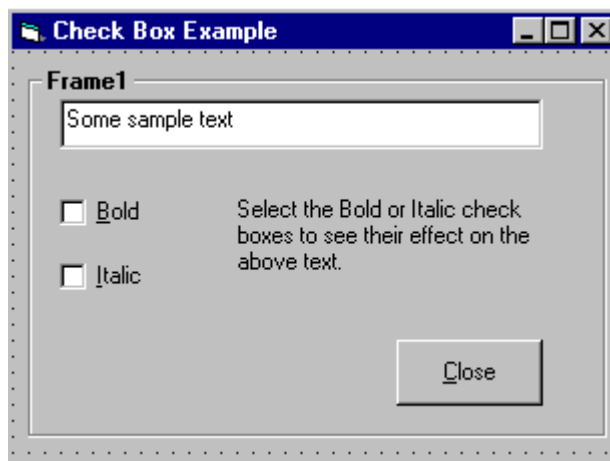
Note: Il peut y avoir des moments où recalculer plus de problèmes de pointe prend trop de temps à chaque changement et ainsi obliger l'utilisateur à entrer toutes les données et ensuite cliquez sur un bouton de réponse pourrait être plus appropriée.

En utilisant l'événement événement GotFocus

Jusqu'à présent, un seul événement a été utilisé par le contrôle, mais cela ne doit pas être le cas! Ajoutez un contrôle StatusBar au bas du formulaire, faire apparaître la fenêtre de code à l'aide , Sélectionnez la première zone de texte (txtNumber1) dans la liste déroulante de gauche, puis sélectionnez l'événement **GotFocus** dans la liste déroulante de droite. Maintenant, quelques instructions de base peut être écrit dans la barre d'état de sorte que lorsque le curseur est dans la zone de texte (la zone de texte a le focus) la barre d'état indique "Entrez le premier nombre". Après avoir terminé ce changement à la deuxième zone de texte et en utilisant le changement même événement GotFocus le texte d'état pour "Entrez un second nombre". Le code pour définir la barre d'état peut être vu ci-dessous.

CheckBoxes

Option bars sont utilisés très souvent dans l'environnement Windows car ils ne peuvent avoir que deux sorties 0 et 1, ces processus de s'habituer à la forme. Dans cet exemple, il sera utilisé pour modifier le texte de la normale à certains gras ou en italique.



```
Private Sub chkBold_Click ()
    Si chkBold.Value = 1 Then 'Si elle est cochée.
```

```
txtDisplay.FontBold = True
Else 'Si ce n'est pas vérifié.
txtDisplay.FontBold = False
End If
End Sub
```

```
Private Sub chkItalic_Click ()
Si chkItalic.Value = 1 Then 'Si elle est cochée.
txtDisplay.FontItalic = True
Else 'Si ce n'est pas vérifié.
txtDisplay.FontItalic = False
End If
End Sub
```

Cet exemple peut être trouvé à "**smamples / PGuide / contrôle / Controls.vbp**» ou téléchargés gratuitement sur la page de téléchargement.

Les cases à cocher peut être activée lors de l'exécution en tapant simplement

```
name.value = 1 '1 Sur, 0 off
```

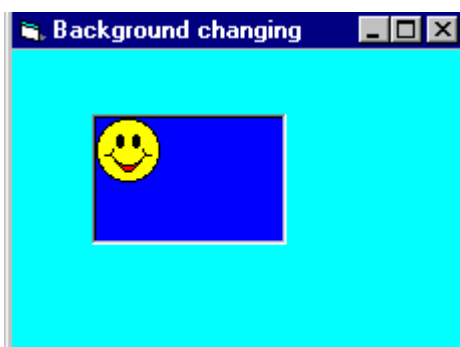
Note: Si vous créez la première image, puis ajoutez les boutons d'option par simple clic sur la boîte à outils et faites glisser le curseur en forme de croix sur le cadre pour créer les contrôles, ils seront fixés au cadre et se déplace avec elle si vous décidez de re-positionner le cadre. Notez, cependant, que si vous créez la première image et double-cliquez sur les contrôles d'écran, puis faites-les glisser du centre de la forme sur le cadre, ils ne seront pas attachés et seront laissées lorsque vous essayez de déplacer le cadre. Essayez ceci.

Notez que lorsque vous exécutez votre application sur l'icône même est chargé en premier (sans doute le presse-papiers, si vous avez créé cette option premier bouton). Vous pouvez modifier l'option qui a le premier foyer, en sélectionnant l'un des autres boutons d'option et en définissant sa propriété tabindex à 1.

Boutons d'option

Changer la couleur de fond

Changer la couleur de fond devient surtout utilisée par freeware, ou le type de programmes que vous générez pour une utilisation avec des bannières ou des annonces, de toute façon il pourrait bien utile parfois. Cet exemple montre une image ordinaire d'un visage souriant, puis-je mettre dans certaines couleurs de fond agréable de faire ressortir davantage.



```
Private Sub Form_Load ()
BackColor = QBColor (* Tour 15)
ForeColor = QBColor (* Tour 10)
Picture1.BackColor = QBColor (* Tour 15)
```





```
Picture1.ForeColor = QBColor (* Tour 10)  
End Sub
```

Les zones de liste


Note:

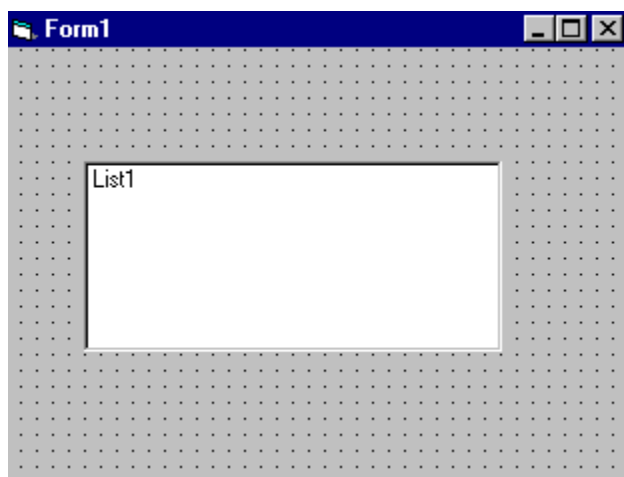
Les zones de liste et zones de liste déroulante sont utilisés pour fournir une liste d'options à l'utilisateur.

La boîte à outils des icônes représentant ces deux contrôles sont  pour zone de liste et  pour la boîte combo.

Une zone de liste est utilisé lorsque l'utilisateur doit être présenté avec un ensemble fixe de sélections (c'est à dire un choix doit être fait uniquement à partir des éléments affichés, il n'ya pas de possibilité de l'utilisateur en tapant une alternative).

Les exemples pourraient être offrant une liste des jours dans une semaine, les modules disponibles dans un catalogue d'option, les destinations de vacances disponibles à partir d'un aéroport ou les types de traitement offert par un institut de beauté.

Pour créer une zone de liste, double-cliquez sur l'icône de boîte à outils . Faites glisser la zone résultant en place sur la forme et la taille qu'il en fonction des données qu'il tiendra. L'image de gauche ci-dessous montre une zone de liste qui a été créé et entreprises sur Form1. Au milieu se trouve le code qui est nécessaire pour charger une sélection de voitures dans la liste. Les données ont été inclus dans la procédure Sub Form_Load de sorte qu'il apparaît lorsque Form1 est chargé. Enfin, l'image de droite montre ce qui apparaît sur le formulaire lorsque l'application est exécutée. Notez que les barres de défilement verticales sont ajoutés automatiquement si la zone de liste n'est pas assez profonde pour afficher tous les choix.



Cependant, si vous ajoutez le code source ci-dessous pour ce projet

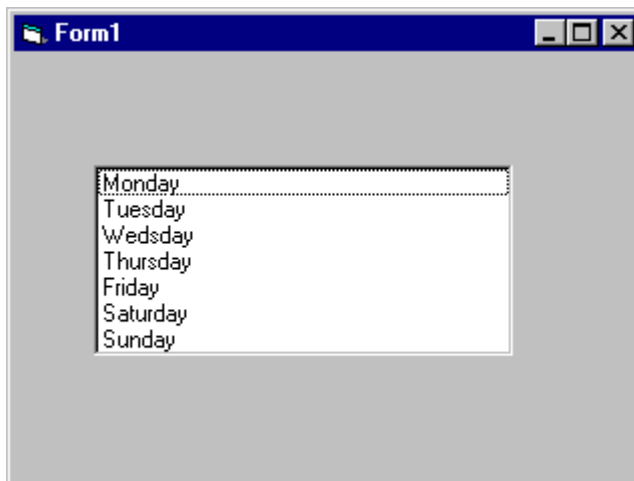
Ajouter à une Lisbox

Private Sub Form_Load ()

```
List1.AddItem "lundi"  
List1.AddItem "mardi"  
List1.AddItem "Wednesday"  
List1.AddItem "jeudi"  
List1.AddItem "Vendredi"  
List1.AddItem "Saturday"
```

```
List1.AddItem "dimanche"  
End Sub
```

Le code source devrait ressembler something comme ceci:



Retrait & Options avancées

Note: Ils apparaissent dans l'ordre où ils ont été tapés si vous avez changé la fenêtre des propriétés en changeant l'ordre de tri = true, puis ils iront dans l'ordre alpaetical. Les éléments de liste peuvent être ajoutés et supprimés toute la liste est considérée comme l'ordre où ils sont dans la pour exapmple si vous voulez supprimer "mardi" vous devez taper

list1.RemoveItem 1

Et pour ajouter à la liste dans un ordre paticular il suffit d'ajouter

list1.additem «My Day», 5

Ce sera ajouté, après le samedi.

Et enfin clairement le type lisbox

List1.clear Ce sera complètement effacer la contence de la listbox.

Note: Les magasins ListCount la propriété du nombre d'éléments dans une liste, de sorte list1.ListCount peut être utilisé pour déterminer le nombre d'éléments de list1 zone de liste.

La propriété ListIndex donne l'indice de l'élément de la liste sélectionnée. Ainsi, la déclaration list1.RemoveItem List1.ListIndex supprime l'élément de liste en surbrillance.

Ajout d'un élément peut être réalisé très proprement en utilisant une boîte de dialogue d'entrée. Essayez ceci:

```
list1.AddItem InputBox ("Entrez un jour», «Ajouter un jour»)
```

Cela ouvrira une boîte de message et vous invite à une nouvelle journée d'entrer dans ce sera alors ajouté à la liste des index à la bottem sauf si vous spécifiez étaient il devrait aller.

Zones de liste déroulante

Combo boîtes sont de trois types (0,1 et 2), définissant leurs propriétés / styles détermine le type. boîtes combo (style 0 et 2) sont un bon choix si l'espace est limité, becaue la liste complète est affichée comme une liste déroulante, il n'occupe pas l'espace écran jusqu'à ce que la flèche du bas est cliqué.

Image de style liste déroulante 0



L'utilisateur peut entrer du texte dans le champ d'édition ou le sélectionner dans la liste des articles en cliquant sur le (détachés) vers le bas flèche vers la droite. La boîte déroulante, la liste est disponible en cliquant sur la flèche vers le bas. Ce type de boîte combo n'a pas de flèche vers le bas car la liste est affichée en tout temps. S'il ya plus d'éléments que peut être affichée dans la taille de la boîte que vous avez dessiné, barres de défilement verticales sont automatiquement ajoutés. Comme pour le type précédent, les utilisateurs peuvent entrer du texte dans le champ d'édition.

Zone de liste déroulante (Style = 2)

Il est un peu déroutant de trouver ce contrôle en vertu de zone de liste déroulante. Cette commande se comporte comme une zone de liste ordinaire, sauf que les choix ne sont pas révélées jusqu'à ce que la flèche vers le bas est cliqué. L'utilisateur peut sélectionner uniquement parmi les choix donnés, il n'ya pas de centre de saisie de texte.

Remarque: les boîtes combo de 0 style et 2 ne peut pas répondre à une double cliquez sur les événements.

Leçon 3

Msgboxes

Les boîtes de message sont utilisés lorsque vous voulez demander à l'utilisateur une question ou d'afficher un message d'erreur (s) et informer l'utilisateur. Il ya six types de boîtes de message ici sont leurs fonctions et ce qu'ils font. Voici la liste de tous les événements msgbox possible

Les boutons affichés dans un message ici Button Layout	Valeur	Description succincte
vbOKOnly	0	Affiche le bouton OK.
vbOKCancel	1	Affiche le bouton OK et Annuler.
vbAbortRetryIgnore	2	Affiche les boutons Abandonner, Réessayer, Ignorer
vbYesNoCancel	3	Affiche Oui, Non et bouton Annuler
vbYesNo	4	Affiche le Oui / Non bouton
vbRetryCancel	5	Affiche les boutons Réessayer et Annuler.

Les icônes dispayed dans la boîte de message sont ici

Icône sur le message	Valeur	Description succincte
vbCritical	16	critique icône de message affiche
vbQuestion	32	Affiche l'icône question
vbExclamation	48	point d'exclamation Affiche
vbInformation	64	icon Affiche des informations

	Valeur	Description succincte
--	--------	-----------------------

Le bouton par défaut affiché dans un formulaire de message Bouton par défaut		
vbDefaultButton1	0	Le bouton 1 est par défaut
vbDefaultButton2	256	Bouton 2 est par défaut
vbDefaultButton3	512	Bouton par défaut est 3

Valeur de retour MsgBox	Valeur de retour	Valeur	Description succincte
vbOk		1	L'utilisateur a cliqué sur OK
vbCancel		2	L'utilisateur a cliqué Annuler
vbAbort		3	L'utilisateur a cliqué Abort
vbRetry		4	L'utilisateur a cliqué Réessayer
vbIgnore		5	Ignorer l'utilisateur a cliqué
VbYes		6	L'utilisateur a cliqué sur Oui
VbNo		7	L'utilisateur a cliqué n

La syntaxe pour l'utilisation de la boîte de message dans MsgBox "TEXT", VALEUR, "TITLE"
Si vous souhaitez utiliser deux ou plusieurs tables il suffit d'ajouter les valeurs. Ainsi, pour imprimer une boîte de message pour dire: «Le périphérique n'a pas été trouvé!" OK & Explication:

1 code source

Private Sub Form_Load ()

MsgBox "Le périphérique n'a pas été trouvé!", 48 ", en-tête"

End Sub

2 Code Source

Private Sub Form_Load ()

MsgBox "Le périphérique n'a pas été trouvé!", VbExclamation, "tête"

End Sub

Vous devriez obtenir l'image ci-dessous ce que vous avez utilisé le code source.



Il s'agit d'un msgbox de base qui, dans ce cas n'a pas été traitée de quelque façon. Le code source suivant affiche un msgbox que vous demandez à un texte spécifique. À titre d'exemple permet de fabriquer un programme de mot de passe de cette boîte de message.

Private Sub Form_Load ()

```
IngBefore = Timer  
Ne  
Straus = InputBox ("Quel est le mot de passe mot de passe est l'exemple", "Mot de passe requis")  
Loop Until Val (Straus) = Exemple  
IngAfter = Timer  
msgbox "Mot de passe incorrect", vbInformation
```

End Sub

Une fois que vous copiez et collez le code source, vous devriez obtenir invité à entrer un mot de passe dans un autre type de msgbox car il comprend le texte. En regardant cet exemple, vous devriez être en mesure de voir comment fonctionne la fonction de boucle pour générer et vérifier la valeur. Tous Valeurs de retour au travail de la même manière pour retourner un objet d'entrée.

Lorsque les applications sont chargées, elles normales pouvaient obtenir un certain paramètre du registre ou fichier présent article va vous montrer comment récupérer une chaîne sur un fichier.

Private Sub Form_Load ()

```
Dim F As Integer, mot de passe As String  
F = FreeFile  
Open App.Path & "\ password.txt" Pour l'entrée Comme F  
F Input #, mot de passe  
Close # F
```

End Sub

Comme vous pouvez le voir sur ce code source le mot de passe est déjà déclaré comme une chaîne. Après ceci est fait n'oubliez pas de fermer le fichier autrement la prochaine fois que vous souhaitez enregistrer ou de lire les fichiers de l'ordinateur vont penser qu'il est utilisé par une autre application et les fenêtres ne vous laissera pas faire quelque chose avec elle. Donc, comme vous pouvez le voir il est **très important** de fermer le fichier

Stockage des informations dans un fichier

programmes FTP souvent stocker des informations dans un fichier comme un nom d'utilisateur et mot de passe ou informations sur l'hôte de la même manière. L'exemple suivant va mettre des informations dans le fichier.

Private Sub Form_Load ()

```
F Dim As Integer, passer As String  
F = FreeFile  
enregistrer txtNew =  
Open App.Path & "\ password.txt" pour la sortie Comme F  
Donnez votre # F, Text1.Text  
Close # F
```

End Sub

Bien que ce soit un peu évidentes, je pense que je devrais l'inclure tout encas je pense différemment à d'autres personnes.

Impression de texte à l'imprimante.

C'est assez facile à faire et il est utilisé dans le bloc-notes, etc ..

Private Sub Form_Load ()

```
Printer.print "L'imprimante permet d'imprimer ce texte"  
Printer.print ""  
Printer.print "Il laissera une ligne ici"  
Printer.print "Il convient d'ajouter la contence de Text1.Text ici:" & Text1.Text & "Comme vous pouvez  
le voir ça marche"  
Printer.print ""  
Printer.EndDoc "Cela indique à l'imprimante qu'il a terminé."
```

End Sub

Tout ce qui appears en position (A) auront imprimé par l'imprimante par défaut printer.print "A". Le enddoc imprimante est utilisée pour indiquer à l'imprimante le travail est terminé, si ce n'est pas ajouté l'imprimante ne peut pas estimer à quelle distance il sera jusqu'à ce qu'il ait fini et quand il a terminé, il pense qu'il has'nt alors assurez-vous d'inclure cette pour éviter toute confusion.

Tableaux de contrôles

Un tableau de commande est une liste de contrôles avec le même nom. Par conséquent, au lieu de l'aide de cinq boutons de commande avec cinq noms distincts, vous pouvez placer un tableau de contrôles bouton de commande sur le formulaire, et ce tableau de contrôle détient cinq boutons de commande. Le tableau de contrôle peut avoir un nom unique, et vous permettra de distinguer le contrôle de l'autre avec un indice. L'une des meilleures raisons d'utiliser un tableau de commande à partir de ce premier contrôle, tous les éléments dans le tableau de commande de prendre sur les valeurs de propriété même, Vous pouvez ensuite modifier ces propriétés qui doivent être changés sans avoir à définir toutes les propriétés pour chaque contrôle individuellement. tableaux de contrôle ont beaucoup en commun avec les tableaux de données. Un tableau de contrôle a un tableau, et de vous distinguer tous les contrôles du tableau de l'autre avec l'indice de base zéro. (La propriété index contient le nombre indice des contrôles). Tous les éléments de contrôle doit être le même type de données. Dès que vous placez un contrôle sur un formulaire qui a le même nom que d'un contrôle existant, Visual Basic permet de s'assurer que vous vous voulez commencer un tableau de contrôles par l'émission du message d'avertissement pour montrer que le contrôle est déjà en cours d'utilisation. Il est utilisé comme un construit dans la sécurité afin que vous n'ayez pas plus de droit d'un contrôle existant en mettant ce quelque part ailleurs sur la même forme. Si vous répondez à la boîte d'avertissement avec un bouton non, Visual Basic utilise un nom de contrôle par défaut pour le contrôle placé.

Non disponible pour le moment Image!

Toutes les procédures cas où l'utilisation de contrôle à partir d'un tableau de contrôle exigent une valeur d'argument spéciale adoptée à eux que le détermine laquelle le contrôle est en cours d'élaboration. Par exemple, si votre application contient une commande unique bouton nommé cmdtotal le click () l'événement commence et se termine comme suit

Privé cmdtotal_click sub ()

End Sub

Toutefois, si vous créez un tableau de contrôle nommé le même nom que précédemment (cmdtotal) il finira comme celui-ci

Privé sous cmdtotal_click (index As Integer)

End Sub

La procédure utilise l'argument d'index comme le numéro d'index de contrôle (l'indice) que l'utilisateur a cliqué, donc si vous souhaitez modifier la propriété de commande cliqué légende boutons à l'intérieur de la cmdtotal_click () les procédures dont vous auriez besoin sont les suivants Cmdtotal (index). caption = "Un nom de légende" La valeur de l'indice est titulaire d'index du bouton de commande du clic de l'utilisateur pour générer les procédures d'événement, afin de toujours répondre à un contrôle adéquat cliqué si vous utilisez Index après le nom du tableau de contrôle.

J'espère que vous allez apprendre ce pendant la leçon 4. :

- Brève introduction à l'usages des bases de données Access
- Base de données d'objets
- Objet Recordset
- Accès aux dossiers
- Recherche dans le RecordSet
- Mise à jour de la base de données
- La suppression et ajout d'enregistrements

mai grâce à Andreas Swensson pour la création de cette page sur des bases de données.

Brève introduction à l'usages des bases de données Access

Ce que je pense que c'est la chose la plus convaincante sur Visual Basic est-il de moyen facile d'accéder et de modifier les bases de données. C'est ce que je pense que vous devriez apprendre par la suite, vous trouverez de nombreuses applications de cette connaissance. Je n'ai presque jamais faire un programme sans l'aide d'une base de données pour le stockage des données.

Il ya plusieurs façons de travailler avec des bases de données en Visual Basic, et je pense que vous avez au moins regardé le contrôle de données. Je ne parle même pas du contrôle de données dans ce texte, car il est si facile à utiliser et trop limitées pour être intéressant pour un développeur professionnel. (Ok, il ya quelques exceptions.)

Ce que je vais vous apprendre à utiliser dans ce texte est DAO (Data Access Objects). Vous vous familiariser avec l'ouverture d'une base de données et la récupération / ajout / suppression / mise à jour des enregistrements de tables. Je vais seulement utiliser une base de données Access (*.mdb) dans mes exemples, car c'est le SGBD le plus utilisé (DataBase Management System) pour les petites applications fait en Visual Basic. Nous allons à la fin de cette leçon ont fait une application de téléphonie simple, mais fonctionnelle, livre.

Ce texte exige une certaine connaissance du langage de programmation Visual Basic et vous devriez être familier avec l'IDE de Visual Basic (Integrated Development Environment).

Base de données d'objets

La première chose que vous devez faire votre demande est d'ouvrir une base de données où sont stockés vos tables. Vous devez déclarer une variable pour stocker votre base de données afin de le faire. Cela se fait avec:

```
Dim dbMyDB As Database
```

Cela vous donne une variable / objet qui peut contenir une référence à votre base de données. Pour ouvrir une base de données Access simple nommée "MyDatabase.mdb", faites ceci:

```
Set dbMyDB = OpenDatabase ("MyDatabase.mdb")
```

Vous devriez vraiment préciser le chemin d'accès complet à la db, mais si votre répertoire courant est le répertoire où est situé la base de données, cela va fonctionner.

Donc, maintenant que vous avez ouvert une base de données. Cela ne vous donne pas toutes les données. Ce que vous devez faire est d'ouvrir une table dans la base de données. Vous n'êtes pas limité à ouvrir une table unique, parfois vous avez deux ou plusieurs tables qui sont liées les unes aux autres et reliés entre eux avec les clés étrangères, et il existe des moyens pour gérer cela. Mais dans ce "Visual Basic - Base de données Primer" Je veux seulement vous montrer comment ouvrir une table unique.

Objet Recordset

Visual Basic utilise un objet appelé RecordSet de réserver votre table. Pour déclarer un tel objet et pour ouvrir la table, procédez comme suit:

```
Dim rsMyRS Comme RecordSet

Set rsMyRS = dbMyDB.OpenRecordSet ("MaTable", dbOpenDynaset)
```

Qu'est-il arrivé là? Eh bien, j'ai déclaré un objet RecordSet et utilisé la méthode de l'objet de base de données de OpenRecordset pour ouvrir une table de type feuille de réponse dynamique. Vous pouvez ouvrir un RecordSet en plusieurs modes. VB fichier d'aide en ligne explique les différents modes et de ce qu'ils ont pour. Le mode feuille de réponse dynamique est le mode que j'utilise la plupart du temps. Il vous donne un RecordSet que vous pouvez ajouter, supprimer et modifier des enregistrements en

Accès aux dossiers

Maintenant que nous avons ouvert une table (dénommé d'enregistrements à partir de maintenant) nous voulons avoir accès aux dossiers en elle. L'objet RecordSet nous permet de passer en elle en utilisant les méthodes MoveFirst, MoveNext, MovePrevious, MoveLast (entre autres). Je vais utiliser certaines de ces pour remplir une zone de liste avec les dossiers de nos RecordSet.

Pour obtenir cet exemple fonctionne, faire une base de données (avec Access) appelé "MyDatabase.mdb" avec la table "MaTable" en elle. Ce tableau doit avoir les champs "ID" de type "Counter" que vous définissez à la clé primaire, le champ "Nom" de type texte et un champ "P hone" de type texte. Ajouter certains enregistrements à elle. Placez une zone de liste sur un formulaire et appelez-le "lstRecords".

```
Dim dbMyDB As Database
Dim rsMyRS Comme RecordSet

Private Sub Form_Load ()

Set dbMyDB = OpenDatabase ("MyDatabase.mdb")
Set rsMyRS = dbMyDB.OpenRecordSet ("MaTable", dbOpenDynaset)

Si ce n'est pas rsMyRS.EOF Puis rsMyRS.MoveFirst
Do While Not rsMyRS.EOF
    rsMyRS lstRecords.AddItem! Nom
    lstRecords.ItemData (lstRecords.NewIndex) = rsMyRS! ID
    rsMyRS.MoveNext
Loop

End Sub
```


Cela rendra la zone de liste remplir avec vos dossiers lors du chargement de formulaire. J'ai introduit quelques nouveaux concepts avec cet exemple. Nous avons tous prêts a porté sur la première partie où nous ouvrons la table. La ligne qui dit `If Not rsMyRS.EOF Then rsMyRS.MoveFirst` indique au programme de se déplacer vers le premier enregistrement dans le cas il ya des registres à tous. L' `EOF` est une propriété booléenne qui est vrai si l'enregistrement en cours est le dernier. Il est également vrai si il n'ya pas de dossiers dans le `RecordSet`.

Ensuite, nous faisons le programme ajouter le champ "Nom" de tous les dossiers de la zone de liste en ajoutant le champ actuel des dossiers "Nom" et de passer à l'enregistrement suivant. Vous demandez un champ d'un `RecordSet` en mettant un `!` entre le nom de l'objet `RecordSet` et la na-moi du champ. Les contrôles boucle `while` pour voir si il ya plus d'enregistrements à ajouter.

Recherche dans le RecordSet

Vous auriez pu me demandais pourquoi je mets la valeur du champ "ID" dans la propriété la zone de liste `ItemData`. Je l'ai fait afin que nous puissions connaître la clé primaire pour tous les enregistrements en vue de rechercher un enregistrement.

Placez une zone de texte quelque part sur la forme et l'appeler "txtPhone". Ensuite, copiez le code ci-dessous pour le projet.

```
Private Sub lstRecords_Click ()  
  
rsMyRS.FindFirst "ID =" & Str (lstRecords.ItemData (lstRecords.ListIndex))  
txtPhone.Text = rsMyRS! Téléphone  
  
End Sub
```

Cela permet d'afficher le numéro de téléphone de la personne sélectionnée lors d'un clic dans la zone de liste. Il utilise la méthode `FindFirst` de l'objet `RecordSet`. Cela prend un paramètre de chaîne qui est comme ce qui est après `WHERE` dans une expression SQL. Vous dites le domaine que vous souhaitez rechercher dans (ici "ID"), puis les critères d'évaluation (ici "=") et enfin la valeur à rechercher (ici le `ItemData` de l'élément sélectionné dans la liste).

Alors que nous avons fait a été de rechercher le dossier avec la valeur "ID" qui a été la même que la propriété `ItemData` de l'élément sélectionné dans la zone de liste. Ensuite, nous montrons la valeur de la "Téléphone" sur le terrain dans la zone de texte.

Mise à jour de la base de données

Vous voudrez probablement être en mesure de mettre à jour une certaine valeur de certains champs où faire la programmation de base de données. Cela se fait avec `Edit` et `Update`. Nous allons essayer de changer la valeur du "Téléphone" sur le terrain en modifiant le texte dans la zone de texte et en cliquant sur un bouton.

Placez un bouton de commande sur le formulaire et nommez-le "cmdUpdate". Ensuite, copiez le code ci-dessous pour le projet.

```
Private Sub cmdUpdate_Click ()  
  
rsMyRS.Edit  
rsMyRS! Téléphone = txtPhone.Text  
rsMyRS.Update  
  
End Sub
```

Pourrait-il être aussi simple que cela? Oui. Cela change Numéros de téléphone de notre personne choisie. Ou pour le dire techniquement: Cela change la valeur du "Téléphone" sur le terrain de notre enregistrement en cours. Imaginez l'enregistrement en cours est un ensemble de boîtes, avec un champ dans chaque case. T-il `Edit` la méthode prend le couvercle toutes les cases et `Update` les met en marche. Lorsque nous écrivons `rsMyRS!Phone = txtPhone.Text` on remplace le contenu du "Téléphone" boîte avec le contenu de la zone de texte.

La suppression et ajout d'enregistrements

Suppression

Suppression d'un enregistrement ne pouvait pas être plus simple. Pour supprimer l'enregistrement en cours que vous venez invoquer la `Delete` de l'objet `RecordSet`. Nous allons mettre cette fonctionnalité dans notre petit projet. Faire une autre bouton de commande appelé "cmdDelete" et le code suivant fera le travail de la suppression de notre personne actuellement sélectionné.

```
Private Sub cmdDelete_Click ()  
  
rsMyRS.Delete  
lstRecords.ListIndex lstRecords.RemoveItem  
  
End Sub
```

Je vais même pas la peine d'expliquer cela plus en détail =). La première instruction supprime l'enregistrement et le second supprime l'entrée de zone de liste.

Ajout

Ajout d'enregistrements est un peu comme `updateing`, sauf que vous utilisez `AddNew` au lieu de `Edit`. Ajoutons encore un bouton de commande à notre demande. Appelons ... `ErRh` ... laissez-moi voir ... oui! "CmdNew" =). Voici le code qui ajoute un nouveau record.

```
Private Sub cmdNew_Click ()  
  
rsMyRS.AddNew  
rsMyRS! Nom = "Une nouvelle personne"  
rsMyRS lstRecords.AddItem! Nom  
lstRecords.ItemData (lstRecords.NewIndex) = rsMyRS! ID  
rsMyRS! Téléphone = "Personne de numéro de téléphone"  
rsMyRS.Update  
  
End Sub
```

Je vais utiliser une analogie pour expliquer cette case. Le `AddNew` méthode prend un ensemble de nouvelles cases et les ajoute à notre `RecordSet`. Nous avons ensuite mettre un peu de nouvelles valeurs en eux et de fermer les paupières avec `Update`. Comme vous pouvez voir que nous n'avons jamais dit tout e précieux pour les "ID", mais comme vous vous en souvenez, il s'agit d'un champ de type "Counter" qui est automatiquement une valeur unique. Le code ajoute également ce nouvel enregistrement à la zone de liste afin que nous serons en mesure de changer le numéro de téléphone de cette personne. Je m'en remets à vous pour ajouter fonctionnalité e e de changer le nom.