Spécification DialogBox

# Objectif

DB permet de créer et configurer des boites de dialogue, on peut donc :

1. Définir le titre d’une boite de dialogue.
2. Définir ses coordonnées et dimensions.
3. Définir sa nature : redimensionnable ou pas et toujours au-dessus ou pas.
4. Insérer différents type d’objet dans la boite de dialogue, actuellement il y a 5 grands type d’objet :
   1. Les libellés qui permettent d’afficher du texte.
   2. Les boutons qui permettent de déclencher une action (script externe).
   3. Les boites à cocher ou bouton radio qui permettent de sélectionner une ou plusieurs options.
   4. Les listes déroulantes qui permettent de sélectionner un élément dans une liste.
   5. Les zones de saisie qui permettent de saisir du texte masqué ou non et des valeurs numériques.
5. Disposer les objets de 3 manières qui sont mixable les unes entre les autres :
   1. Automatique : c’est-à-dire sans se préoccuper de leur positionnement.
   2. Semi-automatique : en se préoccupant de leur espacement horizontale et verticale.
   3. Manuelle : en indiquant leur coordonnées et dimension.
6. Faire intéragir la boite de dialogue avec l’extérieur au moyen de variable d’environnement, de scripts ou de fichiers.

# Moyens

On peut définir une boite de dialogue de 2 façons :

1. Soit en invoquant le programme DialogBox, à la ligne de commande, avec les différentes options (donc, précédées par un tiret) et leurs éventuels arguments, permettants de construire la boite de dialogue.
2. Soit en créant un fichier contenant toutes les options nécessaires à la définition la boite de dialogue, ce fichier sera lu ligne par ligne sachant que chaque début de ligne correspond à la lettre d’option (sans le tiret) telle que passée à la ligne de commande.

Ces 2 façons de faire peuvent être mixées.

L’ordre dans lequel sont placées les options, est important.

Les différentes options possibles sont :

-a : option *"AlwaysOnTop"*, pour indiquer que le dialogue doit être toujours au-dessus des autres fenêtres, par défaut il se comporte normalement.

-r : option *"Resizable"*, afin de rendre le dialogue, retaillable, par défaut il ne l’est pas.

-f nom : option *"File"*, lit le fichier de description de dialogue, indiqué.

-p : option *"Polling"*, scrute les changements sur le fichier de description de dialogue et met à jour en fonction.

-o nom : option *"Outputfile"*, en sortie ou sur évenement (appuie sur un bouton) du dialogue, envoie la valeur des objets dans le fichier indiqué, au format « nomvar="valeur" », par ailleur la valeur des objets est systèmatiquement positionnée dans les variables d’environnement de la routine externe appelée.

-i nom : option *"Inputfile"*, en entrée ou sur retour d’évenement (appuie sur un bouton) du dialogue, lit la valeur des objets, au format « nomvar="valeur" », dans le fichier indiqué, si cette option n’est pas fournie, la sortie standard de la routine externe est scrutée afin de tenter de récupérer ces valeurs toujours au format « nomvar="valeur" ».

-x valnum : option *"XCoordinate"*, si l’option est rencontrée avant toute définition d’objet, elle indique l’origine en X du dialogue, sinon celle du dernier objet déclaré.

-y valnum : option *"YCoordinate"*, même chose pour l’origine en Y.

-w valnum : option *"Width"*, même chose pour la largeur.

-h valnum : option *"Height"*, même chose pour la hauteur.

-t titre : "Title", modifie le titre du dialogue.

-b $nom "libellé", action : "Button", ajoute un bouton avec l’éventuel nom qui servira à l’identifier et son éventuel libellé (s’il ne contient pas d’espace ou de virgule, les guillemets sont facultatives). Au moins un des 2 arguments est nécessaire, si le libellé est manquant, il sera déduit du nom (sans le signe dollar), l’inverse ne sera vrai que dans le cas d’un libellé "OK", "Valider", "Cancel" ou "Annuler" (cf. plus bas). Après une virgule, une action facultative (appel d’un script externe) peut-être fournie. 2 cas de figure particuliers sont à préciser :

1. Si un bouton a comme nom $OK ou comme libellé « OK » ou « Valider », alors il sera le bouton de validation par défaut (DefPushButton) qui s’active quand on appuie sur <Entrer>.
2. Si un bouton a comme nom $cancel ou comme libellé « Cancel » ou « Annuler », alors il sera le bouton d’annulation par défaut qui s’active quand on appuie sur la touche <Escape>.

-c $nom libellé : "", ajoute une boite à cocher (CheckBox). Au moins un des 2 arguments est nécessaire, si le libellé est manquant, il sera déduit du nom (sans le signe dollar), l’inverse n’étant pas vrai. Un comportement supplémentaire de cette option permet de gérer des boutons radio en donnant à plusieurs objet de ce type un nom ayant la même racine mais se différenciant sur le dernier caractére, ils seront ainsi rattachés en un ensemble de bouton radio permettant de n’en cocher qu’un à la fois.

-l $nom "libellé" : "",ajoute un libellé. Au moins un des 2 arguments est nécessaire, si le libellé est manquant, il sera déduit du nom (sans le signe dollar), l’inverse n’étant pas vrai. Les guillemets ne sont nécesaire que si le libellé contient des retours chariot.

-e $nom "valeur de saisie ar défaut" : ajoute une zone de saisie texte. Au moins un des 2 arguments est nécessaire, si le libellé est manquant, il sera déduit du nom (sans le signe dollar), l’inverse n’étant pas vrai. L’espace (' ') ou le souligné ('\_') ont le rôle particulier d’agrandir la largeur interne de l’objet.

-m valtext : "", Ajoute une zone de saisie masqué (mot de passe)

-d valtext[,valtext, ...] : "", Ajoute une drop list (combo).

-s valnum : "", Décale de n pixels vers la droite.

-n : "", Passe à la ligne suivante.

-q action : "", Modifie l'action de sortie du dialogue, le mot clef [Confirm] demande confirmation avant de sortir.

# Comportement

de DB en entrée et sortie