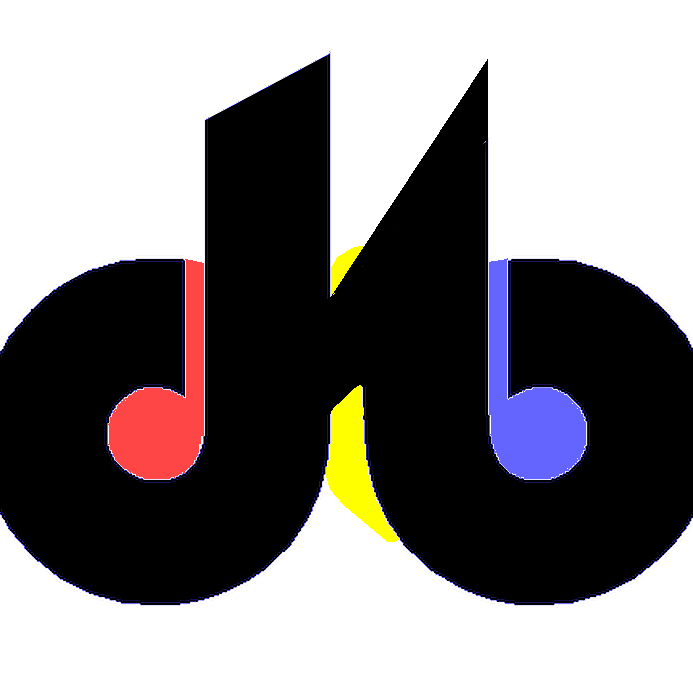
DialogBox



Spécification et fonctionnement

Denis LALANNE

Table des matières

[I. Objectif 1](#_Toc366883277)

[1. Présentation 1](#_Toc366883278)

[2. Moyens 1](#_Toc366883279)

[II. Utilisation 1](#_Toc366883280)

[1. Préalable 1](#_Toc366883281)

[2. Options sans argument 2](#_Toc366883282)

[3. Option avec un argument numérique facultatif 2](#_Toc366883283)

[4. Options avec un argument numérique 2](#_Toc366883284)

[5. Options avec un seul argument texte 2](#_Toc366883285)

[6. Options avec au moins un argument 3](#_Toc366883286)

# Objectif

## Présentation

DialogBox (DB) permet de créer, configurer et faire fonctionner des boites de dialogue.

On peut donc :

1. Définir une boite de dialogue avec son titre, sa nature, son comportement et ses coordonnées et dimensions.
2. Insérer et disposer différents objets dans ce dialogue (textes, boutons, menus …)
3. Faire interagir la boite de dialogue avec son environnement.

## 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 nécessaires.
2. Soit en créant un fichier décrivant les options, une par ligne, nécessaires à la définition et au fonctionnement de la boite de dialogue.

Ces 2 méthodes peuvent être mixées.

# Utilisation

## Préalable

Ici le terme mnémonique est employé pour qualifier la version longue d’une option. Ils peuvent être utilisés aussi bien en majuscule qu’en minuscule.

Chaque description commence par l’option requise, suivie de son mnémonique et enfin l’explication détaillée de son fonctionnement, sous la forme suivante :

***-o / --option-longue****: blablablabla …*

En ligne de commande les options ou mnémoniques sont employés tels quels, suivis de leur éventuels arguments.

Tandis qu’en mode fichier ils doivent se situer exactement à la 1ére colonne d’une ligne, suivi des arguments. Mais sans être précédé par un tiret.

Enfin une description générique peut être donnée qui concernera, selon les indications fournies, plusieurs options.

## Options sans argument

**-a / --AlwaysOnTop** : indique que le dialogue doit être toujours au-dessus des autres fenêtres, par défaut il se comporte normalement, c’est-à-dire qu’il est recouvrable par les autres fenêtres.

**-p / --PollFile** : met à jour le dialogue dès que son fichier de description est modifié.

**-r / --Resizable** : rend le dialogue, retaillable. Par défaut il ne l’est pas.

Remarque : les ascenseurs (scrollbars) du dialogue sont gérés automatiquement, si un ou plusieurs objets se trouvent hors de la zone de visibilité du dialogue alors l’ascenseur concerné apparait automatiquement, afin de permettre d’y accéder.

## Option avec un argument numérique facultatif

**-n / --Newline** : sans argument génère un saut de ligne suffisant pour ne pas toucher l'objet du dessus. Avec argument numérique positif ou négatif, ajuste l'espacement vertical au pixel prêt.

## Options avec un argument numérique

**-s / --Space** : modifie l'espacement horizontal par défaut entre 2 objets, en négatif comme en positif, au pixel prêt.

Remarque : l’utilisation systèmatique de cette option est inutile car un espacement horizontal standard est automatiquement inséré lors de la création de 2 objets contigus. Elle est seulement utile quand on veut modifier cet espacement standard.

**-x / --Xcoordinate** : valeur numérique entière indiquant l'origine en X.

**-y / --Ycoordinate** : valeur numérique entière indiquant l'origine en X.

**-w / --Width** : valeur numérique entière indiquant la largeur.

**-h / --Height** : valeur numérique entière indiquant la hauteur.

Si une de ces 4 options positionnement est rencontrée avant toute définition d’objet, elle s'applique au dialogue, sinon elle s'applique au dernier objet déclaré. L'absence des paramètres X et Y pour le dialogue l’affichera automatiquement au centre de l’écran. Tout objet non repéré est positionné automatiquement, en fonction des espacements horizontaux et verticaux.

## Options avec un seul argument texte

**-t / --Title** : titre du dialogue.

**-f / --File** : nom du fichier de description de dialogue à exécuter.

**-j / --JotFile** : nom du fichier ou seront notés les éventuels avertissements à l'exécution de DialogBox, par défaut c'est la sortir d'erreur standard.

**-o / --OutputFile** : nom du fichier à mettre à jour en sortie de programme ou à) chaque événement (appuie sur un bouton) du dialogue, la valeur d'affichage des objets nommés, est envoyée dans le fichier indiqué, par ailleurs la valeur des objets est systématiquement positionnée dans les variables d’environnement de la routine externe appelée.

**-i / --InputFile** : nom du fichier à lire en entrée ou sur retour d’événement (appuie sur un bouton) du dialogue, la valeur d'affichage des objets nommés, est lue au format «nomvar="valeur"» et ils sont mis à jour. Si cette option n’est pas fournie, la sortie standard de la routine externe est scrutée afin de tenter de récupérer ces valeurs affichées au même format.

Comportement générique relatif au 2 options précédentes :

* S'il n'y a aucun objet nommé dans le dialogue alors il n'y pas de gestion de variable, ni entrée ni en sortie.
* Le format de lecture ou écriture de la valeur d'un objet est : nomvar="valeur".
* Les guillemets ne sont obligatoires que s’il y a un retour chariot dans la valeur. Ou tout autre caractére interprétable dans le contexte de l’objet concerné (cf. plus bas).

## Options avec au moins un argument

Il s’agit ici des options de création d’objet, comprenant éventuellement un nom de variable et un libellé et qui permettent d’insérer des objets d’interfaçage graphique dans le dialogue.

**-l / --Label** : ajoute un libellé.

*Exemple :* --Label "Bonjour  
Comment allez-vous ?  
Très bien, merci !"

**-b / --Button** : ajoute un bouton. La déduction du nom ou de la valeur du bouton ne sera systèmatique que dans le cas d’un nom ou libellé comme : 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 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>.

*Exemple :* -b $MonBouton "Libellé de mon bouton", cmd /c dir & pause

**-c /--CheckBox** : ajoute une boite à cocher (CheckBox). Un comportement supplémentaire de cette option permet de gérer des boutons radio en donnant à plusieurs objet de ce type un nom avec la même racine mais se différenciant sur le dernier caractère, ils seront ainsi regroupés en un ensemble de bouton radio permettant de n’en cocher qu’un à la fois.

*Exemples :*

Checkbox $MaBoiteACocher

--Checkbox $MonBoutonRadio1  
c $MonBoutonRadio2   
-c $MonBoutonRadio3

*Ces 3 déclarations donneront une seule variable d’environnement, nommée $MonBoutonRadio et dont la valeur sera celle du libellé du bouton coché. Ou vide si aucun n’est coché.*

**-e / --Edit** : zone de saisie de texte.

*Exemples :*

--Edit $login "\_\_\_\_\_\_\_\_\_\_"  
-e "Valeur par défaut"

**-m / --MaskedEdit** : zone de saisie masquée.

*Exemple :* -m "\_\_\_\_\_\_\_\_\_\_"

**-u / --nUmEdit** : zone de saisie d'une valeur numérique.

*Exemple :* -u "1922"

**-v / --VerifiedEdit** : zone dont la saisie est contrôlée pour respecter un format prédéfini qui est décrit à l’aide des caractères suivant :

@ : caractère alphabétique.

A : caractère alphabétique en majuscule uniquement.

a : caractère alphabétique en minuscule uniquement.

n : chiffre de 0 à 9.

m : nombre de 01 à 12 (mois)

d : nombre de 01 à 31 (jour du mois)

h : nombre de 00 à 24 (heure sur 24h).

H : nombre de 00 à 12 (heure sur 12h).

M : nombre de 00 à 59 (seconde ou une minute).

S : idem.

\_ : Le souligné indique un caractère quelconque.

\ : Annule la signification des caractères de formatage qui précédent.

Tout autre caractère fourni à l’initialisation servira « d’enrobage » et restera donc inchangé au moment de la saisie.

*Exemples :*

-v "d/m/nnnn" # peut indiquer une date

-v "d/m/2nnn" # peut indiquer une date supérieure au 31 décembre 1999 et inférieure au 01/01/3000

-v "h:s:s" # peut indiquer une heure entre 00:00:00 et 24:59:00

-v "h:m:m" # idem

-v "h\hm\ms\s" # peut indiquer une heure entre 00h00m00s et 24h59m00s

**-d / --DropdownList** : ajoute une liste déroulante dont le contenu doit-être spécifié à la suite, chaque choix étant séparé par une virgule. La valeur de variable d’environnement sera uniquement celle sélectionnée.

*Exemple* : -d "1er choix liste,2éme choix liste,…"

Comportement générique relatif à toutes les options d’objet :

* Au moins un des 2 arguments, nom de variable ou libellé est nécessaire, si le libellé est manquant, il sera déduit du nom (sans le signe dollar), l’inverse n’étant pas systématique.
* Les guillemets ne sont nécessaires que si le libellé contient des retours chariot et/ou tout autre caractère susceptible d'être interprété dans le contexte de l'objet (la virgule pour un bouton, par exemple).
* L'espace ou le souligné peuvent être utilisés pour dimensionner le libellé.
* Le nom de variable fourni pour ces option sert aussi bien à récupérer la valeur d’affichage de l’objet depuis l’environnement appelant qu’à générer une valeur pour le même nom de variable dans un environnement appelé via l’appuie sur un bouton.

*Options inutilisées*

*k*

*g GraphicObject*

*z Zbuffered3DObject*