

NOM

dialog - Afficher des boîtes de dialogue à partir de scripts d'interpréteur de commandes

SYNOPSIS

```
dialog --clear  
dialog --create-rc fichier  
dialog --print-maxsize  
dialog options_communes options_boîte
```

DESCRIPTION

Dialog est un programme qui permet de présenter une grande variété de questions ou d'afficher des messages en utilisant des boîtes de dialogue depuis un script d'interpréteur de commandes. Ces types de boîte de dialogue sont implémentés (bien que tous ne soient pas nécessairement compilés dans **dialog**) :

buildlist, **calendar**, **checklist**, **dselect**, **editbox**, **form**, **fselect**, **gauge**, **infobox**, **inputbox**, **inputmenu**, **menu**, **mixedform**, **mixedgauge**, **msgbox** (message), **passwordbox**, **passwordform**, **pause**, **prgbox**, **programbox**, **progressbox**, **radiolist**, **rangebox**, **tailbox**, **tailboxbg**, **textbox**, **timebox**, **treeview** et **yesno** (oui/non).

Plusieurs boîtes de dialogue peuvent être mises dans un script :

- utiliser l'indicateur **--and-widget** pour forcer **dialog** à traiter le dialogue suivant à moins d'avoir appuyé sur ESC pour annuler, ou ;
- ajouter simplement les indications pour la boîte de dialogue suivante, réalisant ainsi une chaîne. **dialog** cesse l'enchaînement quand le code de retour est différent de zéro, par exemple, Annuler ou Non (voir DIAGNOSTICS).

Certains composants graphiques, par exemple, **checklist**, écrivent le texte sur la sortie de **dialog**. Normalement, c'est la sortie d'erreur standard, mais il existe des options pour changer cela : « **--output-fd** », « **--stderr** » et « **--stdout** ». Aucun texte n'est écrit si le bouton Annuler (ou ESC) est pressé, **dialog** abandonne immédiatement dans ce cas.

OPTIONS

Toutes les options commencent par « **--** » (deux tirets ASCII, pour ceux qui utilisent des systèmes avec une prise en charge de la locale incomplète).

Un « **--** » par lui-même est utilisé comme échappement, c'est-à-dire, l'indication suivante sur la ligne de commande n'est pas traitée comme une option. Cela est différent de [getopt\(1\)](#) qui utilise cette indication pour traiter les indications suivantes comme *paramètres* plutôt que comme *options*.

dialog --title -- --Pas une option

dialog --title Cela -- --titre n'est pas une option

dialog n'utilise pas de *paramètres* et utilise son propre analyseur d'*options*.

Quand une option commune (par exemple, qui n'est pas un composant graphique) est répétée, la dernière option rencontrée est celle utilisée. Les options booléennes sont gérées spécialement de façon à pouvoir être annulées en ajoutant (ou en omettant) un modificateur « **no** » après le « **--** » de début. Par exemple, **--no-shadow** est documentée ici, mais **--shadow** est aussi acceptée.

L'option « **--args** » indique à **dialog** de lister les paramètres de la ligne de commande sur la sortie d'erreur standard. Cela est utile pour le débogage de scripts complexes en utilisant « **--** » et « **--file** » puisque la ligne de commande peut être réécrite lors de leur développement.

L'option « **--file** » indique à **dialog** de lire les paramètres depuis le fichier indiqué.

dialog --file fichier_paramètre

Les blancs non entourés de guillemets droits doubles sont ignorés (utiliser des contre-obliques pour mettre entre guillemets des caractères uniques). Le résultat est inséré dans la ligne de commande, remplaçant « **--file** » et sa valeur d'option. L'interprétation de la ligne de commande continue à partir de ce point. Si *fichier_paramètre* commence avec « **&** », **dialog** interprète le texte qui suit comme un numéro de descripteur de fichier plutôt que comme un nom de fichier.

La plupart des composants graphiques acceptent des paramètres *hauteur* et *largeur* qui peuvent être utilisés pour dimensionner automatiquement le composant graphique pour s'adapter aux valeurs de message multiligne d'*invite* :

- si le paramètre est négatif, **dialog** utilise la taille de l'écran ;
- si le paramètre est zéro, **dialog** utilise la taille minimale du composant graphique pour afficher l'*invite* et les données ;
- autrement, **dialog** utilise la taille indiquée du composant graphique.

Options courantes

La plupart des options communes sont réinitialisées avant de traiter chaque composant graphique.

--ascii-lines

Plutôt que de dessiner des lignes autour des boîtes, dessiner des « **+** » et des « **-** » ASCII aux mêmes emplacements (voir aussi « **--no-lines** »).

Cette option permet un certain contrôle des dimensions de la boîte lors de l'utilisation du dimensionnement automatique (en spécifiant 0 pour la hauteur et la largeur). Elle représente le rapport hauteur/largeur. La valeur par défaut est 9, ce qui signifie une largeur de 9 caractères pour chaque hauteur de ligne.

--backtitle *titre_fond*

Indication d'une chaîne *titre_fond* pour un affichage en arrière-plan en haut de l'écran.

--begin *y x*

Indication de la position du coin supérieur gauche de la boîte de dialogue sur l'écran.

--cancel-label *chaîne*

Remplacement du libellé utilisé pour les boutons « Annulation ».

--clear

Effacement de l'écran du composant graphique, conservant uniquement la couleur d'arrière-plan. Cette option est à utiliser lors d'une combinaison de composants graphiques avec « **--and-widget** » pour effacer le contenu du composant graphique précédent sur l'écran de façon à ce qu'il n'apparaisse pas dans le contenu d'un composant graphique suivant. Cette option est à considérer comme un complément de « **--keep-window** ». Pour comparer les effets, utilisez ce qui suit :
Trois composants graphiques visibles, effet d'escalier, ordre 1,2,3 :

```
dialog \
```

```
        --begin 2 2 --yesno "" 0 0 \
```

```
        --and-widget        --begin 4 4 --yesno "" 0 0 \
```

```
        --and-widget        --begin 6 6 --yesno "" 0 0
```

Seul le dernier composant graphique reste visible :

```
dialog \
```

```
        --clear        --begin 2 2 --yesno "" 0 0 \
```

```
        --and-widget --clear        --begin 4 4 --yesno "" 0 0 \
```

```
        --and-widget        --begin 6 6 --yesno "" 0 0
```

Trois composants graphiques visibles, effet d'escalier, ordre 3,2,1 :

```
dialog \
```

```
--keep-window --begin 2 2 --yesno "" 0 0 \
```

```
--and-widget --keep-window --begin 4 4 --yesno "" 0 0 \
```

```
--and-widget --begin 6 6 --yesno "" 0 0
```

Premier et troisième composants graphiques visibles, effet d'escalier, ordre 3,1 :

```
dialog \
```

```
--keep-window --begin 2 2 --yesno "" 0 0 \
```

```
--and-widget --clear --begin 4 4 --yesno "" 0 0 \
```

```
--and-widget --begin 6 6 --yesno "" 0 0
```

Remarque : pour restaurer les couleurs d'origine de console et réintégrer le curseur après la fin du programme **dialog**, utiliser la commande `clear(1)`. Inversement, pour effacer l'écran et réintégrer le curseur au bas gauche après la fin du programme **dialog**, utiliser l'option **--erase-on-exit**.

--color-mode *code*

Extension de l'option **--colors** pour le contenu de programbox, tailbox et textbox. Utiliser le code « 2 » pour cette fonctionnalité.

--colors

Interprétation des séquences « \Z » imbriquées dans la boîte de dialogue par le caractère qui suit pour indiquer à **dialog** de définir les attributs de couleurs et de vidéo :

- 0 à 7 sont les codes ANSI de couleur utilisés dans curses, respectivement : noir, rouge, vert, jaune, magenta, cyan et blanc.
- le gras est défini par « b » et annulé par « B » ;
- l'inversion est définie par « r » et annulée par « R » ;
- le soulignement est défini par « u » et annulé par « U » ;
- les réglages sont cumulatifs, par exemple, « \Zb\Z1 » rend le texte qui suit en rouge gras (peut-être vif) ;
- les réglages sont restaurés avec « \Zn ».

Demande à **dialog** de diviser les données pour les boîtes radio/check et les menus lors des occurrences de la chaîne indiquée et d'aligner ces données en colonnes.

--cr-wrap

Interprétation des caractères de nouvelle ligne imbriqués dans le texte du dialogue comme de nouvelles lignes sur l'écran. Autrement, **dialog** repliera les lignes où cela est nécessaire pour rentrer dans la boîte de texte.

Même s'il est possible de contrôler les changements de ligne avec cela, **dialog** replie toute ligne trop longue pour tenir dans la boîte. Sans l'option `cr-wrap`, la disposition du texte peut être formatée joliment dans le code source du script sans affecter la manière dont il apparaîtra dans la boîte de dialogue.

La fonctionnalité *cr-wrap* est mise en œuvre sous les conditions suivantes :

- la chaîne contient « \n » et l'option **--no-nl-expand** n'est pas utilisée ;
- ou l'option **--trim** est utilisée.

Pour plus d'informations, consulter **Options d'espaces blancs**.

--create-rc *fichier*

Si **dialog** gère la configuration d'environnement d'exécution, cette option peut être utilisée pour copier un fichier d'exemple de configuration dans le fichier indiqué par *fichier*.

--cursor-off-label

Placement du curseur du terminal à la fin du bouton plutôt que sur le premier caractère du libellé du bouton. Cela est utile pour réduire la confusion visuelle quand la couleur du bouton se différencie peu de celle du son libellé.

--date-format *format*

Si l'hôte prend en charge **strftime**, cette option permet de préciser le format de la date affichée dans le composant graphique **--calendar**. Le moment du jour (heure, minute, seconde) est au temps local actuel.

--defaultno

Réglage de la valeur par défaut de la boîte **yes/no** à **No**. De même, traiter le bouton par défaut des composants graphiques qui fournissent « Ok » et « Annuler » comme un *Cancel*. Si les options « **--no-cancel** » ou « **--visit-items** » sont fournies, elles outrepassent cette option, rendant le bouton par défaut toujours « Yes » (de même pour « OK »).

--default-button *chaîne*

Réglage du bouton (présélectionné) par défaut dans un composant graphique. En présélectionnant un bouton, un script rend possible l'utilisation par l'utilisateur d'une simple pression sur *Entrée* pour exécuter le dialogue avec une interaction minimale.

La valeur de l'option est le nom du bouton : *ok*, *yes*, *cancel*, *no*, *help* ou *extra*.

Normalement, le premier bouton de chaque composant graphique est celui par défaut. Le premier bouton affiché est déterminé par le composant graphique de concert avec les options « **--no-ok** » et « **--no-cancel** ».

Si cette option n'est pas indiquée, aucun bouton n'est assigné par

défaut.

--default-item *chaîne*

Réglage de l'élément par défaut dans une boîte checklist, form ou menu. Normalement le premier item de la boîte est celui par défaut.

--erase-on-exit

Lorsque **dialog** quitte, supprimer le composant graphique de dialogue en remettant l'écran entier à sa couleur native d'arrière-plan et en plaçant le curseur du terminal au coin bas de gauche.

--exit-label *chaîne*

Remplacement du libellé utilisé pour les boutons « EXIT ».

--extra-button

Afficher un bouton supplémentaire entre les boutons « OK » et « Annuler ».

Le bouton supplémentaire apparaît entre « Yes » et « No » pour le composant graphique *yesno*

--extra-label *chaîne*

Remplacement du libellé utilisé pour les boutons « Extra ». Note : pour les composants graphiques inputmenu, la valeur par défaut est « Rename ».

--help

Affichage d'un message d'aide sur la sortie standard et quitter. Le message d'aide est aussi affiché si aucune option n'est précisée ou si une option inconnue est indiquée.

--help-button

Affichage d'un bouton d'aide après les boutons « OK » et « Annuler » dans des boîtes qui ont une liste d'items étiquetés (c'est-à-dire une boîte checklist, radiolist, menu ou treeview).

Le bouton d'aide apparaît après « Yes » et « No » pour le composant graphique *yesno*.

Lors de l'abandon, le code de retour indique que le bouton Help a été appuyé. **dialog** écrit aussi un message sur sa sortie après l'étiquette « HELP » :

- si « **--item-help** » est aussi indiquée, le texte d'aide de l'item est écrit ;
- autrement, l'étiquette de l'item (le premier champ) est écrit.

L'option **--help-tags** peut être utilisée et/ou la variable d'environnement `DIALOG_ITEM_HELP` peut être réglée pour modifier ces messages et le code de retour.

Cette option peut être utilisée pour les autres composants graphiques qui ont un bouton « OK », que le bouton « Annuler » soit utilisé ou non. Le code de retour et la sortie ne sont pas traités spécialement pour les autres composants graphiques, le bouton d'aide est seulement un bouton supplémentaire.

--help-label *chaîne*

Remplacement du libellé des boutons « Help ».

--help-status

Si le bouton d'aide est sélectionné, les informations de checklist, radiolist ou form seront affichées après l'information « HELP » d'item-help. Cela peut être utile pour reconstruire l'état d'une checklist.

après le traitement d'une requête d'aide.

--help-tags

Modification des messages écrits lors de l'abandon pour **--help-button** en les modifiant toujours à la simple étiquette de l'item. Cette option ne modifie pas le code de retour d'abandon.

--hfile *fichier*

Affichage du fichier indiqué dans une textbox lorsque l'utilisateur appuie sur F1.

--hline *chaîne*

Afficher la chaîne indiquée centrée au bas du composant graphique. Scrollable windows have an indicator near the left side of the widget, and a corresponding space is reserved on the right side. This limits the amount of text which can be displayed to 9 columns less than the window width. If insufficient space is available, no text is shown.

--ignore

Ignorance des options non reconnues par **dialog**. Certaines options bien connues telles que « **--icon** » sont quand même ignorées, mais c'est le meilleur choix pour la compatibilité avec d'autres implémentations.

--input-fd *desc_de_fichier*

Lecture de la saisie au clavier pour le descripteur de fichier indiqué. La plupart des scripts de **dialog** lisent l'entrée standard, mais le composant graphique gauge lit un tube (qui est toujours l'entrée standard). Certaines configurations ne fonctionnent pas correctement quand **dialog** essaie de réouvrir le terminal. Cette option est à utiliser (avec un réarrangement approprié des descripteurs de fichier) si un script doit fonctionner dans ce type d'environnement.

--insecure

Rendu du composant graphique de mot de passe plus convivial, mais moins sécurisé, en affichant un astérisque pour chaque caractère.

--iso-week

Définition du point de départ du numéro de semaine affiché dans l'option « **--calendar** » selon la norme ISO-8601, qui démarre la numérotation avec la première semaine qui inclut un jeudi en janvier.

--item-help

Interprétation des données d'étiquette pour les boîtes checklist, radiolist et menu en ajoutant une colonne affichée au bas de l'écran pour l'item actuellement sélectionné.

--keep-tite

Lorsque construit avec **ncurses**, **dialog** vérifie normalement s'il est exécuté dans un **xterm** et, dans ce cas, essaie de supprimer les chaînes d'initialisation qui pourraient le faire basculer vers un écran de remplacement. Basculer entre l'écran normal et celui de remplacement est gênant visuellement lors d'un script qui exécute **dialog** plusieurs fois. Cette option est à utiliser pour permettre à **dialog** d'utiliser ces chaînes d'initialisation.

--keep-window

Normalement quand **dialog** crée plusieurs composants graphiques **tailboxbg** connectés par « **--and-widget** », il efface l'ancien composant graphique de l'écran par superposition. Cette option est à utiliser pour supprimer cette superposition.

Lors de l'abandon, **dialog** superpose tous les composants graphiques.

ont été marqués avec « **--keep-window** », même s'ils ne sont pas des composants graphiques **tailboxbg**. Cela fait qu'ils sont superposés en ordre inverse. Consulter les explications de l'option « **--clear** » pour des exemples.

--last-key

Lors de l'abandon, rapport de la dernière touche appuyée. C'est le code de touche de cursors plutôt qu'un symbole ou un caractère littéral, et ce rapport n'est fait que pour les touches liées à une action. Cette option peut être utilisée par des scripts pour faire la distinction entre deux touches liées à la même action.

--max-input *size*

Limitation des chaînes d'entrée à une taille donnée. Si elle n'est pas indiquée, la limite est 2048.

--no-cancel

Suppression du bouton « Annuler » dans les modes de boîte checklist, inputbox et menu. Un script peut toujours tester si l'utilisateur appuie sur la touche ESC pour empêcher l'abandon.

--no-collapse

Normalement, **dialog** convertit les tabulations en espaces et réduit plusieurs espaces à une seule pour le texte qui est affiché dans une boîte de message, etc. Cette option est à utiliser pour désactiver cette fonctionnalité. Remarque : **dialog** repliera toujours le texte soumis aux options « **--cr-wrap** » et « **--trim** ».

La fonctionnalité *no-collapse* est mise en œuvre sous les conditions suivantes :

- la chaîne contient « \n » et l'option **--no-nl-expand** n'est pas utilisée ;
- ou l'option **--trim** n'est pas utilisée ;

Pour plus d'informations, consulter **Options d'espaces blancs**.

--no-hot-list

Indication pour **dialog** de supprimer la fonctionnalité de raccourci de clavier pour les listes, par exemple, checkbox, menu.

Normalement, le premier caractère en capitale d'une entrée de liste sera mis en évidence, et la saisie de ce caractère déplacera le focus sur cette entrée. Cette option supprime la mise en évidence et le mouvement.

Les raccourcis clavier pour « OK », « Annuler », etc ne sont pas affectés.

--no-items

Certains composants graphiques (checklist, inputmenu, radiolist, menu) affichent une liste avec deux colonnes (« étiquette » et « item », c'est-à-dire « description »). Cette option indique à **dialog** de lire des lignes plus courtes en omettant la partie « item » de la liste. Cela est occasionnellement utile, par exemple, si les étiquettes fournissent assez d'informations.

Voir aussi **--no-tags**. Si les deux options sont indiquées, cette dernière est ignorée.

--no-kill

Demande à **dialog** de mettre la boîte **tailboxbg** en arrière-plan,

affichant son identifiant de processus sur la sortie de **dialog**. SIGHUP est désactivé pour le processus d'arrière-plan.

--no-label *chaine*

Outrepassement du libellé utilisé pour les boutons « No ».

--no-lines

Plutôt que de dessiner des lignes autour de la boîte, dessiner des espaces au même endroit. Voir aussi « **--ascii-lines** ».

--no-mouse

Désactivation de la souris.

--no-nl-expand

Pas de conversion des sous-chaines « \n » du texte de message/invite en nouvelles lignes littérales.

La fonctionnalité *no-nl-expand* est utilisée seulement si la chaine contient « \n » de façon à avoir quelque chose à convertir.

Pour plus d'informations, consulter **Options d'espaces blancs**.

--no-ok

Suppression du bouton « OK » pour ne pas l'afficher. Un script peut toujours tester si l'utilisateur presse la touche « Entrée » pour accepter les données :

.

la touche « Entrée » est toujours gérée comme bouton « OK » quand l'option **--no-ok** est utilisée, c'est-à-dire que par défaut elle est liée à la touche virtuelle *LEAVE* ;

quand l'option **--no-ok** n'est pas utilisée, la touche *Tab* peut être utilisée pour déplacer le curseur à travers les champs et les boutons du composant graphique. Dans ce cas, la touche *Entrée* active le bouton sur lequel le curseur est positionné ;

.

pour pouvoir activer un bouton lors de l'utilisation de l'option **--no-ok**, une autre touche *LEAVE* virtuelle existe qui active le bouton en cours. Par défaut, **^D** (EOF) est lié à cette touche.

--no-shadow

Suppression de l'ombrage à la droite et au bas de chaque boîte de dialogue.

--no-tags

Certains composants graphiques (checklist, inputmenu, radiolist, menu) affichent une liste avec deux colonnes (étiquette et description).

L'étiquette est utile pour des scripts mais peut ne pas aider

l'utilisateur. L'option **--no-tags** (de *Xdialog*) peut être utilisée pour supprimer de l'affichage la colonne d'étiquettes. Au contraire de l'option **--no-items**, cela n'affecte pas les données lues à partir du script.

Xdialog n'affiche pas la colonne d'étiquettes pour les composants graphiques analogues *buildlist* et *treeview*. **dialog** fait de même.

Normalement, **dialog** permet un déplacement rapide entre les entrées de la liste affichée en établissant une correspondance à un seul caractère avec le premier caractère de l'étiquette. Quand l'option **--no-tags** est

indiquée, **dialog** établit la correspondance avec le premier caractère de la description. Dans tous les cas, le caractère utilisé comme correspondance est mis en évidence.

--ok-label *chaîne*

Outrepassement du libellé utilisé pour les boutons « OK ».

--output-fd *desc_de_fichier*

Écriture de la sortie dans le descripteur de fichier indiqué. La plupart des scripts de **dialog** écrivent sur la sortie standard d'erreur, mais les messages d'erreur peuvent être écrits ici selon l'indication du script.

--separator *chaîne*

--output-separator *chaîne*

Spécification d'une chaîne séparant la sortie lors d'une sortie de **dialog** à partir de checklists, à la place d'une nouvelle ligne (pour **--separate-output**) ou d'une espace. Cette option s'applique à d'autres composants graphiques tels que form et editbox qui normalement utilisent une nouvelle ligne.

--print-maxsize

Affichage de la taille maximale des boîtes de dialogue , c'est-à-dire la taille de l'écran, pour la sortie de **dialog**. Cela peut être utilisé seul sans autres options.

--print-size

Affichage de la taille de chaque boîte de dialogue sur la sortie de **dialog** lors de l'initialisation de la boîte.

--print-text-only *chaîne* [*hauteur* [*largeur*]]

Affichage de la chaîne telle qu'elle serait repliée dans une boîte de message dans la sortie de **dialog**.

À cause des valeurs facultatives par défaut de *hauteur* et *largeur* à zéro, si elles sont omises, **dialog** définit automatiquement la taille à celle de l'écran.

--print-text-size *chaîne* [*hauteur* [*largeur*]]

Affichage de la taille de la chaîne telle qu'elle serait repliée dans une boîte de message dans la sortie de **dialog** sous la forme :

hauteur largeur

À cause des paramètres facultatifs par défaut de *hauteur* et *largeur* à zéro, s'ils sont omis, **dialog** définit automatiquement la taille à celle de l'écran.

--print-version

Affichage de la version de **dialog** dans la sortie de **dialog**. Cette option peut être utilisée seule sans autres options et ne provoque pas que **dialog** quitte de lui-même.

--quoted

Normalement, **dialog** met entre guillemets droits les chaînes renvoyées par les checklist ainsi que le texte d'item-help. Cette option est à utiliser pour mettre entre guillemets tous les résultats de chaîne si nécessaire (c'est-à-dire si la chaîne contient un caractère d'espace blanc ou de guillemet droit simple ou double).

--reorder

Par défaut, le composant graphique buildlist utilise le même ordre pour la sortie (droite) que pour l'entrée (gauche). Cette option est à utiliser pour indiquer à **dialog** l'ordre dans lequel un utilisateur ajoute des sélections dans la liste de sortie.

--scrollbar

Pour les composants graphiques possédant une liste déroulante de données, dessin d'une barre de défilement sur la marge droite, non manipulable avec la souris.

--separate-output

Pour certains composants graphiques (buildlist, checklist, treeview), production du résultat une ligne à la fois, sans guillemets. Cela facilite l'analyse par un autre programme.

--separate-widget *chaine*

Spécification d'une chaîne séparant la sortie dans la sortie de **dialog** de chaque composant graphique. Cette option est utilisée pour simplifier l'analyse du résultat d'une boîte de dialogue ayant plusieurs composants graphiques. Si cette option n'est pas spécifiée, par défaut la chaîne de séparation est le caractère de tabulation.

--single-quoted

Utilisation de guillemets droits simples si c'est nécessaire (pas de guillemets autrement) pour la sortie du texte d'une checklist ou de item-help.

Si cette option n'est pas spécifiée, **dialog** peut utiliser des guillemets droits doubles autour de chaque item. Dans tous les cas, **dialog** ajoute des contre-obliques pour que la sortie soit utilisable dans des scripts d'interpréteur de commandes.

Des guillemets droits simples peuvent être nécessaires si la chaîne contient des caractères d'espace blanc ou des guillemets droits doubles ou simples.

--size-err

Vérification de la taille résultante de la boîte de dialogue avant de l'utiliser et son affichage si elle est supérieure à celle de l'écran (cette option est obsolète puisque tous les appels new-window sont vérifiés).

--sleep *secs*

Mise en veille (délai) du nombre de secondes indiqués après le traitement d'une boîte de dialogue.

--stderr

Sortie directe sur la sortie standard d'erreur. C'est le comportement par défaut puisque curses écrit normalement les mises à jour d'écran sur la sortie standard.

--stdout

Sortie directe sur la sortie standard. Cette option est fournie pour la compatibilité avec Xdialog. Cependant, l'utiliser dans des scripts portables n'est pas recommandé puisque curses écrit normalement les mises à jour d'écran sur la sortie standard. Si cette option est utilisée, **dialog** essaie de réouvrir le terminal pour écrire sur l'écran. En fonction de la plateforme et de l'environnement, cela peut échouer.

--tab-correct

Conversion de chaque caractère de tabulation en une ou plusieurs espaces (pour le composant graphique **textbox**, autrement une seule espace). Sinon les tabulations sont rendues selon l'interprétation de la bibliothèque curses. L'option **--no-collapse** désactive cette expansion.

--tab-len *n*

Spécification du nombre d'espaces que le caractère de tabulation occupe si l'option « **--tab-correct** » est indiquée. La valeur par défaut est huit. Cette option est seulement effective pour le composant graphique **textbox**.

--time-format *format*

Si l'hôte prend en charge **strftime**, cette option permet de préciser le format de l'heure affichée par le composant graphique **--timebox**. Les valeurs du jour, du mois et de l'année sont dans ce cas celles locales.

--timeout *secs*

Délai en secondes en cas de non-réponse de l'utilisateur. Un délai de zéro seconde est ignoré.

Normalement, le délai fait qu'un caractère ESC est entré dans le composant graphique en cours et l'annule. D'autres composants graphiques peuvent toujours être sur l'écran et ne sont pas annulés. Définir la variable d'environnement **DIALOG_TIMEOUT** pour indiquer à **dialog** de quitter directement, c'est-à-dire effacement de tous les composants graphiques sur l'écran.

Cette option est ignorée par le composant graphique « **--pause** ». Elle est aussi outrepassée si l'option d'arrière-plan « **--tailboxbg** » est utilisée pour définir plusieurs composants graphiques simultanés.

--title *titre*

Spécification d'une chaîne de titre à afficher au haut de la boîte de dialogue.

--trace *fichier*

Journalisation des paramètres de ligne de commande, des frappes de touche et d'autres informations dans le fichier indiqué. Si **dialog** lit un fichier de configuration, cela est aussi journalisé. La capture de l'entrée dans le composant graphique *gauge* est journalisée. Ctrl/T est utilisé pour journaliser une image de la fenêtre de dialogue en cours. Le programme **dialog** gère quelques paramètres de ligne de commande de façon spéciale et les supprime de la liste de paramètres au fur et à mesure qu'ils sont traités. Par exemple, si la première option est **--trace**, ce qui est traité (et supprimé) avant que **dialog** initialise l'affichage.

--week-start *jour*

Spécification du premier jour de la semaine utilisé par l'option « **--calendar** ». Le paramètre *jour* peut être :

- un nombre (0 à 6, dimanche à samedi utilisant POSIX) ;
- la valeur spéciale « locale » (cela fonctionne avec les systèmes utilisant la glibc, fournissant une extension à la commande **locale**, la valeur **first_weekday**) ;
- une chaîne de correspondance d'abréviation de jour de la semaine affiché dans le composant graphique **calendar**, par exemple, « Mo » pour « lundi ».

--trim

Élimination des blancs de début du texte de message et élimination des répétitions de nouvelle ligne et de blancs.

La fonctionnalité *trim* est mise en œuvre aux conditions suivantes :

- la chaîne ne contient pas de « \n » ou ;
- l'option **--no-nl-expand** est utilisée.

Pour plus d'informations, consulter **Options d'espaces blancs**.

Voir aussi les options « **--cr-wrap** » et « **--no-collapse** ».

--version

Affichage de la version de **dialog** sur la sortie standard et abandon.

Voir aussi l'option « **--print-version** ».

--visit-items

Modification du parcours par tabulation des checklist, radiolist, menubox et inputmenu pour inclure la liste des items sous forme d'un des états. Cette option est utile comme aide visuelle, c'est-à-dire que la position du curseur aide quelques utilisateurs.

Lorsque cette option est spécifiée, le curseur est initialement placé sur la liste. Les abréviations (la première lettre de l'étiquette) s'appliquent à la liste d'items. Lors d'un parcours par tabulation jusqu'à la rangée de boutons, les abréviations s'appliquent aux boutons.

--yes-label *chaîne*

Outrepassement du libellé pour les boutons « Yes ».

Options de boîte

Toutes les boîtes de dialogue ont au moins trois paramètres :

texte

la légende ou le contenu de la boîte ;

hauteur

la hauteur de la boîte de dialogue ;

largeur

la largeur de la boîte de dialogue.

Les autres paramètres dépendent du type de la boîte de dialogue.

--buildlist *texte hauteur largeur hauteur_liste [étiquette item état] ...*

Une boîte de dialogue **buildlist** affiche deux listes l'une à côté de l'autre. La liste de gauche affiche les items non sélectionnés, la liste de droite ceux sélectionnés. Lorsque les items sont sélectionnés ou désélectionnés, ils changent de liste.

Un retour chariot ou le bouton « OK » permettent d'accepter la valeur actuelle dans la fenêtre choisie et de quitter. Les résultats sont écrits en utilisant l'ordre affiché dans la fenêtre choisie.

La valeur initiale on/off de chaque entrée est indiquée par *état*.

La boîte de dialogue se comporte comme un **menu**, utilisant l'option **--visit-items** pour contrôler si le curseur est autorisé à parcourir les listes directement :

- si l'option **--visit-items** n'est pas précisée, le parcours avec Tab utilise deux états (OK/Annuler) ;
- si l'option **--visit-items** est spécifiée, le parcours avec Tab utilise quatre états (Gauche/droite//OK/Annuler).

Que l'option **--visit-items** soit spécifiée ou non, il est possible de modifier la mise en évidence en utilisant les touches par défaut :

« ^ » (colonne gauche) et « \$ » (colonne droite).

Lors de l'abandon, une liste de chaînes d'*étiquette* de ces entrées qui sont activées est affichée sur la sortie de **dialog**.

Si l'option « **--separate-output** » n'est pas indiquée, les chaînes seront entre guillemets droits, si nécessaire, pour simplifier leurs séparations dans des scripts. Par défaut, ce sont des guillemets droits doubles. Consulter l'option « **--single-quoted** » qui modifie le comportement de la mise entre guillemets.

--calendar *texte hauteur largeur jour mois année*

Une boîte de dialogue **calendar** affiche le mois, le jour et l'année dans des fenêtres ajustables séparées. Si les valeurs pour le jour, le mois ou l'année sont absentes ou négatives, les valeurs correspondant à la date courante sont utilisées. Il est possible d'augmenter ou de diminuer toutes ces valeurs en utilisant les flèches gauche, haut, bas ou droit. Il est possible d'utiliser les touches h, j, k et l de style vi pour parcourir la grille du mois et d'utiliser les touches Tab et Shift+Tab pour changer de fenêtre. Si l'année indiquée est zéro, la date courante est utilisée comme valeur initiale.

Lors de l'abandon, la date est affichée sous la forme jour/mois/année.

L'option **--date-format** permet de passer outre ce format.

--checklist *texte hauteur largeur hauteur_liste [étiquette item état] ...*

Une boîte de dialogue **checklist** est similaire à une boîte de dialogue **menu**. Plusieurs entrées sont présentées sous la forme d'un menu. Une autre différence est qu'il est possible d'indiquer quelles entrées sont sélectionnées en établissant leur *état* à *on*. Au lieu de choisir une seule entrée, chaque entrée peut être sélectionnée ou désélectionnée. La valeur initiale on/off de l'état de chaque entrée est spécifiée par *état*.

Lors de l'abandon, une liste de chaînes d'*étiquette* de ces entrées qui sont activées est affichée sur la sortie de **dialog**.

Si l'option « **--separate-output** » n'est pas spécifiée, les chaînes seront entre guillemets droits si nécessaire pour que les scripts puissent les séparer simplement. Par défaut, ce sont des guillemets droits doubles. Consulter l'option « **--single-quoted** » qui modifie le comportement de la mise entre guillemets.

--dselect *chemin_fichier hauteur largeur*

La boîte de dialogue de sélection de répertoire affiche une fenêtre text-entry pour saisir un nom de répertoire et au-dessus une fenêtre avec les noms de répertoire.

Ici, *chemin_fichier* peut être un chemin de fichier auquel cas la fenêtre de répertoires affiche le contenu du chemin et la fenêtre text-entry contient le répertoire présélectionné.

Les touches Tab et de flèches permettent de changer de fenêtre. Dans la fenêtre de répertoires, les touches de flèche haut et bas permettent de

parcourir la sélection en cours. La barre d'espace permet de copier le répertoire sélectionné dans la fenêtre text-entry.

La saisie de n'importe quel caractère imprimable déplace le focus sur la fenêtre text-entry, saisissant ce caractère et parcourant la fenêtre de répertoires à la correspondance de caractère la plus proche.

Un retour chariot ou le bouton « OK » sont à utiliser pour accepter la valeur en cours dans la fenêtre text-entry et pour quitter.

Lors de l'abandon, le contenu de la fenêtre text-entry est écrit sur la sortie de **dialog**.

--editbox *chemin_fichier hauteur largeur*

La boîte de dialogue edit-box affiche une copie du fichier. Il est possible de l'éditer en utilisant les touches *Retour_arrière*, *Suppr* et les touches de curseur pour corriger les erreurs de saisie. Les touches page suivante et page précédente sont aussi reconnues. Au contraire de **--inputbox**, la touche Tab doit être utilisée pour sélectionner les boutons « OK » ou « Annuler » pour clore le dialogue. Appuyer sur la touche « Entrée » dans la boîte coupe la ligne correspondante.

Lors de l'abandon, le contenu de la fenêtre d'édition est écrite dans la sortie de **dialog**.

--form *texte haut larg haut_form [libellé y x item y x larg_form larg_entrée]*

...

La boîte de dialogue **form** affiche un formulaire constitué de libellés et de champs qui sont positionnés sur une fenêtre déroulante aux coordonnées données par le script. La valeur des champs *largeur_form* et *largeur_entrée* indiquent leur taille possible. La première définit la largeur du champ sélectionné tandis que la dernière définit la largeur possible des données entrées dans le champ.

- Si *larg_form* est zéro, le champ correspondant ne peut être altéré et le contenu du champ détermine la largeur affichée.
- Si *larg_form* est négative, le champ correspondant ne peut être altéré et la valeur mise en positif de *larg_form* est utilisée pour la largeur d'affichage.
- Si *larg_entrée* est zéro, elle est définie à *larg_form*.

Les flèches haut/bas (ou Crtl/N ou Crtl/P) permettent de naviguer dans les champs et la touche Tab de changer de fenêtre.

Lors de l'abandon, le contenu des champs de formulaire est écrit sur la sortie de **dialog**. Le texte utilisé pour remplir les champs non éditables (*largeur_form* est zéro ou négative) n'y est pas écrit.

--fselect *chemin_fichier hauteur largeur*

La boîte de dialogue **dialog** de sélection de fichier affiche une fenêtre text-entry dans laquelle il est possible de saisir un nom de fichier (ou de répertoire), et au-dessus deux fenêtres pour les répertoires et les noms de fichier.

Ici *chemin_fichier* peut être un chemin, auquel cas les fenêtres de fichiers et de répertoires affichent le contenu du chemin et la fenêtre text-entry contient le nom de fichier présélectionné.

Les touches Tab et de flèche permettent de changer de fenêtre. Dans les fenêtres de répertoires et de fichiers, les touches bas/haut permettent

de faire défiler la sélection en cours. La touche espace permet de copier la sélection en cours dans la fenêtre text-entry.

La saisie de n'importe quel caractère imprimable déplace le focus sur la fenêtre text-entry, en saisissant ce caractère et faisant défiler les fenêtres de répertoires et fichiers à la plus proche correspondance.

La frappe du caractère espace force **dialog** à compléter le nom en cours jusqu'au point où il peut y avoir une correspondance à plus d'une entrée.

Un retour chariot ou le bouton « OK » sont à utiliser pour accepter la valeur en cours dans la fenêtre text-entry et pour quitter.

Lors de l'abandon, le contenu de la fenêtre text-entry est écrit sur la sortie de **dialog**.

--gauge *texte hauteur largeur [pourcentage]*

Une boîte de dialogue **gauge** affiche une barre de progression en bas de la boîte. La barre indique le pourcentage. Les nouveaux pourcentages sont lus depuis l'entrée standard, un entier par ligne. La barre est mise à jour pour refléter chaque nouveau pourcentage. Si l'entrée standard lit la chaîne « XXX », la première ligne qui suit est prise comme un entier de pourcentage, puis les lignes suivantes avec « XXX » sont utilisées pour une nouvelle invite. La jauge s'arrête lorsque EOF est rencontré sur l'entrée standard.

La valeur *pourcentage* indique le pourcentage initial à montrer dans la barre. Si non spécifié, c'est zéro.

Lors de l'abandon, aucun texte n'est écrit sur la sortie de **dialog**. Ce composant graphique n'accepte aucune entrée, donc son code de retour est toujours OK.

--infobox *texte hauteur largeur*

Une boîte de dialogue **info** est basiquement une boîte de dialogue **message**. Cependant, dans ce cas, **dialog** abandonnera immédiatement après l'affichage du message à l'utilisateur. L'écran n'est pas effacé lorsque **dialog** quitte, de façon que le message demeure sur l'écran jusqu'à ce que le script appelant l'efface. Cette boîte de dialogue est utile pour informer l'utilisateur que des opérations sont en cours et peuvent nécessiter un certain temps.

Lors de l'abandon, aucun texte n'est écrit sur l'entrée standard de **dialog**. Un code de retour OK est renvoyé.

--inputbox *texte hauteur largeur [init]*

Cette boîte est utile pour poser une question qui nécessite une réponse de l'utilisateur sous forme d'une chaîne. Si *init* est indiqué, il sera utilisé comme valeur initiale de la chaîne. En saisissant la réponse, il est possible d'employer les touches *Retour_arrière* et *Suppr* et celles de curseur pour corriger les erreurs de frappe. Si la chaîne saisie est trop longue pour la boîte de dialogue, le champ de saisie sera déroulé horizontalement.

Lors de l'abandon, la chaîne d'entrée sera écrite dans la sortie de **dialog**.

--inputmenu *texte hauteur largeur hauteur_menu [étiquette item] ...*

Une boîte de dialogue **inputmenu** est très similaire à une boîte de dialogue **menu** ordinaire. Il y a peu de différences entre elles :

1. Les entrées ne sont pas automatiquement centrées mais ajustées à partir de la gauche ;
2. Un bouton supplémentaire (appelé *Renommer*) est impliqué pour renommer l'item en cours lorsqu'il est pressé ;
3. Il est possible de renommer l'entrée en cours en appuyant sur le bouton *Renommer*. Alors **dialog** écrira ce qui suit sur la sortie de **dialog** :


```
RENAMED <étiquette> <item>
```

--menu *texte hauteur largeur hauteur_menu [étiquette item] ...*

Comme son nom l'indique, une boîte de dialogue **menu** est une boîte qui peut être utilisée pour présenter une liste de choix sous la forme d'un menu pour que l'utilisateur fasse une sélection. Les choix sont affichés dans l'ordre donné. Chaque entrée de menu consiste en une chaîne *étiquette* et une chaîne *item*. L'*étiquette* donne un nom à l'item pour le distinguer des autres dans le menu. L'*item* est une description courte de l'option que l'entrée représente. L'utilisateur peut se déplacer en utilisant les touches de curseur, la première lettre de l'*étiquette* en tant que raccourci de clavier ou l'une des touches 1 à 9. Des entrées de hauteur *hauteur_menu* sont affichées dans le menu en même temps, mais le menu sera déroulant s'il existe plus d'entrées pour cette hauteur.

Lors de l'abandon, l'*étiquette* de l'entrée de menu choisie sera écrite sur la sortie de **dialog**. Si l'option « **--help-button** » est précisée, le texte d'aide correspondant sera écrit si l'utilisateur utilise le bouton d'aide.

--mixedform *texte haut larg haut_form [libellé y x item y x larg_form larg_ent type_ent] ...*

La boîte de dialogue **mixedform** affiche un formulaire constitué de libellés et champs ressemblant beaucoup à la boîte de dialogue **--form**. Elle diffère par l'ajout d'un paramètre de type de champ à chaque description de champ. Chaque bit dans le type indique un attribut du champ :

- 1 caché, par exemple, un champ de mot de passe ;
- 2 lecture seule, par exemple, un libellé.

--mixedgauge *texte hauteur largeur pourcent [étiquette1 item1] ...*

Une boîte de dialogue **mixedgauge** affiche une barre de progression au bas de la boîte. Elle indique le pourcentage.

Elle affiche aussi une liste de valeurs d'*étiquette* et d'*item* au haut de la boîte. Consulter [dialog\(3\)](#) pour les valeurs d'étiquette.

Le *texte* est affiché comme une légende entre la liste et la barre de progression. La valeur *pourcent* indique le pourcentage initial affiché

dans la barre.

Aucune disposition n'est prévue pour lire les données depuis l'entrée standard comme cela est fait pour **--gauge**

Lors de l'abandon, aucun texte n'est écrit sur la sortie de **dialog**. Ce composant graphique n'accepte aucune entrée, donc son code de retour est toujours OK.

--msgbox *texte hauteur largeur*

Une boîte de dialogue **message** est très similaire à une boîte **yes/no**. La seule différence entre elles est que la boîte de dialogue **message** possède un seul bouton **OK**. Cette boîte de dialogue peut être utilisée pour afficher n'importe quel message. Après sa lecture, l'utilisateur peut appuyer sur la touche *Entrée* de façon que **dialog** abandonne et que le script d'interpréteur appelant puisse continuer ses opérations.

Si le message est trop important pour l'espace, **dialog** peut permettre de le dérouler, pourvu que l'implémentation sous-jacente de curses le puisse. Dans ce cas, un pourcentage est affiché au bas du composant graphique.

Lors de l'abandon, aucun texte n'est écrit sur la sortie de **dialog**.

Seul un bouton « OK » est fourni, mais un code de retour ESC peut être renvoyé.

--pause *texte hauteur largeur secondes*

Une boîte de dialogue **pause** affiche une barre de progression au bas de l'écran. Elle indique le nombre de secondes restantes avant la fin de la pause. La pause se termine quand le délai est atteint ou quand l'utilisateur appuie sur le bouton « OK » (état OK), le bouton Annuler ou la touche Echap.

--passwordbox *texte hauteur largeur [init]*

Un boîte de dialogue **password** est similaire à une **inputbox**, excepté que le texte saisi par l'utilisateur n'apparaît pas. Cela est utile pour une invite de mot de passe ou d'autres informations sensibles.

Attention, si quelque chose est indiqué dans « init », elle apparaîtra dans la table de processus du système aux fouineurs occasionnels. Il est aussi perturbant pour les utilisateurs de fournir ces informations avec un mot de passe par défaut qu'ils ne peuvent voir. Pour ces raisons, la non-utilisation de « init » est fortement préconisée.

Consulter « **--insecure** » si vous ne vous préoccupez pas de votre mot de passe.

Lors de l'abandon, la chaîne d'entrée sera écrite dans la sortie de **dialog**.

--passwordform *texte haut larg haut_form [libellé y x item y x larg_form larg_entrée] ...*

C'est identique à **--form** excepté que tous les champs textuels sont traités comme des composants graphiques **password** plutôt que comme des composants graphiques **inputbox**.

--prgbox *texte commande hauteur largeur*

--prgbox *commande hauteur largeur*

Une **prgbox** est très similaire à une **programbox**.

La boîte de dialogue est utilisée pour afficher la sortie de la commande spécifiée comme argument de **prgbox**.

Après la fin de l'exécution des commandes, l'utilisateur peut appuyer

sur la touche *Entrée* de façon que **dialog** quitte et que le script appelant continue ses opérations.

Si quatre paramètres sont fournis, le texte est affiché sous le titre, délinéé du contenu défilant de fichier. Si trois paramètres sont fournis, ce texte est omis.

--programbox *texte hauteur largeur*

--programbox *hauteur largeur*

Une boîte de dialogue **programbox** est très similaire à une boîte **progressbox**. Leur seule différence est que la boîte de dialogue **programbox** affiche un bouton « OK » (mais seulement après la fin de l'exécution de la commande).

Cette boîte de dialogue est utilisée pour afficher la sortie d'une commande. Après la fin de l'exécution de la commande, l'utilisateur peut appuyer sur la touche *Entrée* de façon que **dialog** abandonne et que le script appelant continue ses opérations.

Si trois paramètres sont fournis, le texte est affiché sous le titre, délinéé du contenu défilant de fichier. Si deux paramètres sont fournis, ce texte est omis.

--progressbox *texte hauteur largeur*

--progressbox *hauteur largeur*

Une boîte de dialogue **progressbox** est similaire à une boîte **tailbox**, excepté que :

- a) plutôt que d'afficher le contenu d'un fichier, elle affiche la sortie tubée d'une commande et ;
- b) elle abandonne lorsque la fin du fichier est atteinte (il n'y a pas de bouton « OK »).

Si trois paramètres sont fournis, le texte est affiché sous le titre, délinéé du contenu défilant de fichier. Si deux paramètres sont fournis, ce texte est omis.

--radiolist *texte hauteur largeur hauteur_liste [étiquette item état] ...*

Une boîte de dialogue **radiolist** est similaire à une boîte **menu**. La seule différence est que l'entrée sélectionnée est indiquée en définissant son *état* à *on*.

Lors de l'abandon, l'étiquette de l'item sélectionné est écrite dans la sortie de **dialog**.

--rangebox *texte hauteur largeur val_mini val_max val_défaut*

Cette boîte de dialogue permet à l'utilisateur de choisir parmi une plage de valeurs, par exemple, en utilisant un curseur de défilement. La boîte de dialogue affiche la valeur actuelle dans une barre de défilement (comme la boîte de dialogue *gauge*). Les touches Tab et de flèches déplacent le curseur entre les boutons et la valeur. Quand le curseur est sur la valeur, celle-ci peut être éditée en utilisant :

un mouvement de curseur droite/gauche pour sélectionner un chiffre à modifier ;

les touches +/-

pour augmenter ou diminuer le chiffre par pas de un ;

les touches 0 à 9

Quelques touches agissent aussi dans toutes les positions du curseur :

Début/Fin

réglage de la valeur au maximum ou au minimum ;

Page haut/Page bas

incrémentation de la valeur de façon que le curseur de défilement se déplace d'une colonne.

--tailbox *fichier hauteur largeur*

Affichage du texte d'un fichier dans une boîte de dialogue, de la même façon que la commande « tail -f ». Défilement gauche/droite avec les touches « h » et « l » dans le style de vi ou avec les touches de flèche. « 0 » réinitialise le défilement.

Lors de l'abandon, aucun texte n'est écrit sur la sortie de **dialog**. Seul un bouton « OK » est fourni, mais un code de retour ESC peut être renvoyé.

--tailboxbg *fichier hauteur largeur*

Affichage du texte d'un fichier dans une boîte de dialogue comme une tâche en arrière-plan, de la même façon que la commande « tail -f & ». Défilement gauche/droite avec les touches « h » et « l » dans le style de vi ou avec les touches de flèche. « 0 » réinitialise le défilement. **dialog** traite la tâche d'arrière-plan de manière spéciale si d'autres composants graphiques sont présents (**--and-widget**) sur l'écran en même temps. Jusqu'à leur fermeture (par exemple, avec « OK »), **dialog** exécute tous les composants graphiques **tailboxbg** dans le même processus, interrogeant continuellement pour les mises à jour. La touche Tab peut être utilisée pour aller d'un composant graphique à l'autre sur l'écran et les fermer individuellement, par exemple, en appuyant sur *Entrée*. Une fois que les composants graphiques non-**tailboxbg** sont fermés, **dialog** crée une copie de lui-même en arrière-plan et écrit son identifiant de processus si l'option « **--no-kill** » est fournie.

Lors de l'abandon, aucun texte n'est écrit sur la sortie de **dialog**. Un bouton « EXIT » est seulement fourni comme entrée, mais un code de retour ESC peut être renvoyé.

Remarque : les anciennes versions de **dialog** fourchaient immédiatement et essayaient de mettre à jour l'écran individuellement. En plus d'être mauvais pour les performances, cela n'était pas pratique. Certains anciens scripts peuvent ne pas fonctionner correctement avec le schéma d'interrogation continue.

--textbox *fichier hauteur largeur*.

Une boîte de dialogue **textbox** permet d'afficher le contenu d'un fichier texte dans une boîte de dialogue. C'est comme un simple afficheur de fichier texte. L'utilisateur peut se déplacer dans le fichier en utilisant le curseur, les touches page haut/bas et les touches Début/Fin disponibles sur la plupart des claviers. Si les lignes sont trop longues pour pouvoir être affichées dans la boîte, les touches Droite/Gauche peuvent être utilisées pour faire défiler le texte horizontalement. Les touches de style vi, h, j, k et l peuvent être

utilisées au lieu des touches de curseur, et B et N au lieu des touches du déplacement par page. Les touches de style vi « k » et « j » ou les touches fléchées peuvent être utilisées pour le défilement vertical, et les touches de style vi « h » et « l » ou les touches fléchées pour le défilement horizontal. Un « 0 » réinitialise le défilement gauche/droite. Pour plus de commodité, le style vi de recherche avant ou arrière est aussi utilisable.

Lors de l'abandon, aucun texte n'est écrit sur la sortie de **dialog**. Un bouton « EXIT » est seulement fourni comme entrée, mais un code de retour ESC peut être renvoyé.

--timebox *texte hauteur [largeur heure minute seconde]*

Une boîte de dialogue est affichée permettant de sélectionner l'heure, les minutes et les secondes. Si les valeurs d'heure, de minute ou de seconde sont absentes ou négatives, les valeurs correspondantes du moment actuel sont utilisées. Ces valeurs peuvent être incrémentées ou décrémentées en utilisant les flèches haut, bas, gauche ou droite. Les touches Tab ou Shift+Tab permettent de changer de fenêtre.

Lors de l'abandon, le résultat est écrit sous la forme heure:minute:seconde. Ce format peut être outrepassé en utilisant l'option **--time-format**.

--treeview *texte hauteur largeur hauteur_liste [étiquette item état profondeur] ...*

Affichage arborescent des données. Chaque groupe de données contient une étiquette, le texte à afficher pour l'item, son état (« on » ou « off ») et la profondeur de l'item dans l'arbre.

Un seul item peut être sélectionné (comme pour **radiolist**). L'étiquette n'est pas affichée.

Lors de l'abandon, l'étiquette de l'item sélectionné est écrite dans la sortie de **dialog**.

--yesno *texte hauteur largeur*

Une boîte de dialogue **yesno** de taille *hauteur* lignes par *largeur* colonnes est affichée. La chaîne indiquée par *texte* est affichée dans la boîte de dialogue. Si cette chaîne est trop longue pour tenir sur une ligne, elle sera automatiquement divisée en plusieurs lignes aux emplacements appropriés. La chaîne *texte* peut aussi contenir des sous-chaînes `\n` ou des caractères de nouvelle ligne ``\n'` pour contrôler les ruptures de ligne explicitement. Cette boîte de dialogue est utile pour poser des questions qui nécessitent une réponse de l'utilisateur par oui ou non. La boîte de dialogue a un bouton **Yes** et un bouton **No** que l'utilisateur peut sélectionner à l'aide de la touche *Tab*.

Lors de l'abandon, aucun texte n'est écrit dans la sortie de **dialog**. En plus des codes de retour « Yes » et « No » (voir DIAGNOSTICS), un état de sortie ESC peut être renvoyé.

Les codes utilisés pour « Yes » et « No » correspondent à ceux utilisés pour « OK » et « Annuler ». Aucune distinction n'est faite en interne.

Options obsolètes

--beep

Cette option était utilisée par le `cdialog` originel pour produire un

bip sonore quand des processus distincts du composant graphique `tailbox` rafraichissaient l'écran.

--beep-after

Bip après que l'utilisateur a abandonné un composant graphique en appuyant sur un des boutons.

Options d'espaces blancs

Ces options peuvent être utilisées pour transformer les espaces blancs (espace, tabulation, nouvelle ligne) lorsque **dialog** lit le script :

--cr-wrap, **--no-collapse**, **--no-nl-expand** et **--trim**

Les options ne sont pas indépendantes :

- **dialog** vérifie si le script contient au moins un « `\n` » et (sauf si **--no-nl-expand** est indiqué) ignorera les options **--no-collapse** et **--trim** ;
- après la vérification pour « `\n` » et l'option **--no-nl-expand**, **dialog** prend en charge l'option **--trim** ;

si l'option **--trim** entre en compte, alors **dialog** ignore l'option **--no-collapse**. Il change les séquences de tabulations, d'espaces (et de fins de ligne sauf si **--cr-wrap** est indiqué) en une espace unique ;

.

si les cas « `\n` » ou **--trim** n'entrent en compte, **dialog** recherche l'option **--no-collapse** pour décider ou non de réduire les séquences de tabulations et d'espaces en une espace unique.

Dans ce cas, **dialog** ignore l'option **--cr-wrap** et ne modifie pas les fins de ligne.

En prenant en compte ces dépendances, voici une table résumant le comportement pour les diverses combinaisons d'options. La table suppose que le script contient au moins un « `\n` » quand l'option **--no-nl-expand** n'est pas indiquée.

cr-	no-	no-	trim	Résultat
wrap	collapse	nl- expand		
non	non	non	non	Conversion tabulation en espace. Conversion fin de ligne en espace. Conversion « <code>\n</code> » en fin de ligne.
non	non	non	oui	Conversion tabulation en espace. Conversion fin de ligne en espace. Conversion « <code>\n</code> » en fin de ligne.
non	non	oui	non	Conversion tabulation en espace. Pas de conversion de fin de ligne en espace. Conversion de plusieurs espaces en une. Affichage des « <code>\n</code> » littéralement.

non	non	oui	oui	Conversion tabulation en espace. Conversion de plusieurs espaces en une. Conversion fin de ligne en espace. Affichage des « \n » littéralement.
non	oui	non	non	Conversion fin de ligne en espace. Conversion « \n » en fin de ligne.
non	oui	non	oui	Conversion fin de ligne en espace. Conversion « \n » en fin de ligne.
non	oui	oui	non	Non conversion de fin de ligne en espace. Non réduction de plusieurs blancs. Affichage des « \n » littéralement.
non	oui	oui	oui	Conversion plusieurs espaces en une seule. Conversion de fin de ligne en espace. Affichage des « \n » littéralement.
oui	non	non	non	Conversion tabulation en espace. Repli aux fins de ligne. Conversion « \n » en fin de ligne.
oui	non	non	oui	Conversion tabulation en espace. Repli aux fins de ligne. Conversion « \n » en fin de ligne.
oui	non	oui	non	Conversion tabulation en espace. Pas de conversion de fin de ligne en espace. Conversion de plusieurs espaces en une. Affichage des « \n » littéralement.
oui	non	oui	oui	Conversion tabulation en espace. Conversion plusieurs espaces en une seule. Repli aux fins de ligne. Affichage des « \n » littéralement.
oui	oui	non	non	Repli aux fins de ligne. Conversion « \n » en fin de ligne.
oui	oui	non	oui	Repli aux fins de ligne. Conversion « \n » en fin de ligne.
oui	oui	oui	non	Non conversion de fin de ligne en espace. Non réduction de plusieurs blancs. Affichage des « \n » littéralement.
oui	oui	oui	oui	Conversion plusieurs espaces en une seule. Repli aux fins de ligne. Affichage des « \n » littéralement.

CONFIGURATION D'ENVIRONNEMENT D'EXÉCUTION

1. Créer un exemple de fichier de configuration en saisissant :

```
dialog --create-rc fichier
```

Au démarrage, **dialog** détermine les réglages à utiliser comme suit :

- a) si la variable d'environnement **DIALOGRC** est définie, sa valeur détermine le nom du fichier de configuration ;
- b) si le fichier du (a) n'est pas trouvé, utilisation du fichier `$HOME/.dialogrc` comme fichier de configuration ;
- c) si le fichier du (b) n'est pas trouvé, essai d'utilisation du fichier GLOBALRC lors de la compilation, c'est-à-dire `/etc/dialogrc` ;
- d) si le fichier dans (c) n'est pas trouvé, utilisation des paramètres compilés par défaut.

3. Éditer l'exemple de fichier de configuration et copier-le à un emplacement pouvant être trouvé par **dialog** comme indiqué dans la deuxième étape ci-dessus.

RACCOURCIS CLAVIER

Il est possible d'outrepasser ou d'ajouter des raccourcis de clavier dans **dialog** en les ajoutant dans le fichier de configuration. La commande **bindkey** de **dialog** mappe chaque touche à son codage interne.

`bindkey composant_graphique touche_curses touche_dialog`

Le nom de *composant_graphique* peut être « `*` » (tous les composants graphiques) ou des composants graphiques particuliers tels que **textbox**. Les mappages d'un composant graphique particulier outrepassent les mappages « `*` ». Les mappages définis par l'utilisateur outrepassent les mappages internes de **dialog**.

Les *touche_curses* peuvent être exprimées de différentes façons :

- Cela peut être n'importe quel nom dérivé de *curses.h*, par exemple, « `HELP` » de « `KEY_HELP` ».
- **dialog** reconnaît aussi les caractères de contrôle ANSI tels que « `^A` », « `^?` », ainsi que les contrôles C1 tels que « `~A` » et « `~?` ».
- finalement, **dialog** permet les protections par contre-obliques comme dans C. Cela peut être des valeurs octales de caractère telles que « `\033` » (le caractère d'échappement ASCII), ou les caractères listés dans cette table :

Protégé Réel

<code>\b</code>	espace arrière
<code>\f</code>	saut de page (« form feed ») ;
<code>\n</code>	fin de ligne (nouvelle ligne)

```

\r      retour chariot
\s      espace
\t      tabulation
\^      « ^ » (caret)
\?      « ? » (point d'interrogation)
\\      « \ » (contre-oblique)

```

Les noms de code de touche internes de **dialog** correspondent au type **DLG_KEYS_ENUM** dans **dlg_keys.h**, par exemple, « HELP » de « DLGK_HELP ».

Noms de composant graphique

Certains composants graphiques (tel formbox) ont une zone où les champs peuvent être édités. Ceux-ci sont gérés dans une sous-fenêtre du composant graphique, et peuvent avoir des raccourcis de clavier différents de ceux du composant graphique principal parce que les sous-fenêtres sont enregistrées sous un nom différent.

<i>Composant graphique</i>	<i>Nom de fenêtre</i>	<i>Nom de sous-fenêtre</i>
calendar	calendar	
checklist	checklist	
editbox	editbox	editbox2
form	formbox	formfield
fselect	fselect	fselect2
inputbox	inputbox	inputbox2
menu	menubox	menu
msgbox	msgbox	
pause	pause	
progressbox	progressbox	
radiolist	radiolist	
tailbox	tailbox	
textbox	textbox	searchbox
timebox	timebox	
yesno	yesno	

Certains composants graphiques sont réellement d'autres composants graphiques, utilisant des réglages internes pour une modification de comportement. Ceux-ci utilisent le même nom de composant graphique que le véritable composant graphique :

<i>Composant graphique</i>	<i>Composant graphique réel</i>
dselect	fselect
infobox	msgbox
inputmenu	menu
mixedform	form

passwordbox	inputbox
passwordform	form
prgbox	progressbox
programbox	progressbox
tailboxbg	tailbox

Liaisons internes

Cette page de manuel ne liste pas les raccourcis de clavier pour chaque composant graphique parce que des informations détaillées peuvent être obtenues en exécutant **dialog**. Si l'option **--trace** est indiquée, **dialog** écrit les informations de raccourci pour chaque composant graphique lorsqu'il est déclaré.

Quelques raccourcis sont construits en interne, indépendamment de composants graphiques particuliers :

<i>Touches</i>	<i>But</i>
Ctrl-I	déplacement par tabulation avant, par exemple, avec --tailboxbg
Ctrl-L	rafraichissement de l'écran
Ctrl-T	copie de l'écran dans le fichier --trace
Ctrl-V	suppression des touches spéciales pour l'octet d'entrée suivant
DLGK_FIELD_NEXT	déplacement par tabulation avant, comme Ctrl-I
DLGK_FIELD_PREV	déplacement par tabulation arrière, comme Shift-Tab
DLGK_HELPFILE	affichage du fichier d'aide spécifié avec --hfile
KEY_BTAB	déplacement par tabulation arrière, p. ex., avec --tailbox

Exemple

Normalement, **dialog** utilise des touches différentes pour naviguer entre les boutons et la partie édition d'un dialogue par rapport à la navigation dans la partie édition. C'est-à-dire que la touche Tab (et Shift-Tab) parcourt les boutons (ou les boutons et la partie édition) tandis que les touches de flèches parcourent les champs de la partie édition. La touche Tab est aussi reconnue comme cas spécial pour parcourir les composants graphiques, par exemple, quand plusieurs composants graphiques **tailboxbg** sont employés.

Certains utilisateurs peuvent souhaiter utiliser la même touche pour parcourir la partie édition et les boutons. Le composant graphique **form** a été écrit pour prendre en charge cette sorte de redéfinition de touches en ajoutant un groupe spécial dans **dlgk_keys.h** pour « form » (gauche/droite/suivant/précédent). Voici un exemple de mappage montrant comment faire :

```
bindkey formfield TAB form_NEXT bindkey formbox TAB form_NEXT bindkey
formfield BTAB form_prev bindkey formbox BTAB form_prev
```

Ce type de redéfinition ne devrait pas être utile pour d'autres composants graphiques, par exemple, `calendar`, à cause du nombre potentiellement important de champs à parcourir.

CODE DE RETOUR

Le code de retour est susceptible d'être outrepassé par les variables d'environnement. Les valeurs par défaut et les valeurs correspondantes de variable d'environnement pouvant outrepasser ce code sont :

0	si les boutons YES ou OK sont pressés (<code>DIALOG_OK</code>) ;
1	si les boutons No ou Cancel sont pressés (<code>DIALOG_CANCEL</code>) ;
2	si le bouton Help est pressé (<code>DIALOG_HELP</code>), sauf dans le cas décrit ci-après à propos de <code>DIALOG_ITEM_HELP</code> ;
3	si le bouton Extra est pressé (<code>DIALOG_EXTRA</code>) ;
4	si le bouton Help est pressé, l'option --item-help est définie et la variable d'environnement <code>DIALOG_ITEM_HELP</code> est réglée à 4.
Bien que tous les codes de retour puissent être outrepassés en utilisant des variables d'environnement, ce cas spécial a été introduit en 2004 pour simplifier la compatibilité. dialog utilise <code>DIALOG_ITEM_HELP</code> (4) en interne, mais à moins que la variable d'environnement soit aussi définie, il change cela en <code>DIALOG_HELP</code> (2) lors de l'abandon ;	
5	si un délai expire et que la variable DIALOG_TIMEOUT est définie à 5 ;
-1	si une erreur se produit dans dialog (<code>DIALOG_ERROR</code>) ou que dialog abandonne parce que la touche <i>ESC</i> (<code>DIALOG_ESC</code>) est pressée.

ENVIRONNEMENT

DIALOGOPTS

Cette variable est à définir pour appliquer toutes les options communes à chaque composant graphique. La plupart de ces options sont réinitialisées avant de traiter chaque composant graphique. Si ces options sont définies dans cette variable d'environnement, elles sont appliquées dans l'état de **dialog** après la réinitialisation. Comme dans l'option « **--file** », les guillemets droits doubles et les contre-obliques sont interprétés.

DIALOGRC

Cette variable est à définir pour indiquer le nom du fichier de configuration à utiliser.

DIALOG_CANCEL

DIALOG_ERROR

DIALOG_ESC

DIALOG_EXTRA

DIALOG_HELP

DIALOG_ITEM_HELP

DIALOG_TIMEOUT

DIALOG_OK

Toutes ces variables sont à définir pour modifier le code de retour de :

- Cancel (1) ;
- error (-1) ;
- ESC (255) ;
- Extra (3) ;
- Help (2) ;
- Help avec **--item-help** (2) ;
- Timeout (5), ou ;
- OK (0).

Normalement, les scripts d'interpréteur de commandes ne peuvent faire la distinction entre -1 et 255

DIALOG_TTY

Cette variable est à régler à « 1 » pour fournir la compatibilité avec les anciennes versions de **dialog**, qui assume que si le script redirige la sortie standard, l'option « **--stdout** » est indiquée.

FICHIERS

\$HOME/.dialogrc

fichier de configuration par défaut.

PORTABILITÉ

dialog fonctionne avec curses X/Open. Cependant, certaines implémentations ont des lacunes :

- Curses d'HPUX (et peut-être d'autres) n'ouvre pas le terminal proprement pour la fonction *newterm*. Cela interfère avec l'option **--input-fd** de **dialog** en empêchant les touches de curseur et des séquences d'échappement similaires d'être reconnues.
- Curses de NetBSD 5.1 prend en charge incomplètement les caractères larges. **dialog** pourra être construit, mais tous les exemples ne fonctionnent pas.

pas affichés correctement.

COMPATIBILITÉ

Vous voulez peut-être écrire des scripts qui fonctionnent avec d'autres « clones » de **dialog**

Dialog originel

En premier, considérons le programme **dialog** « originel » (versions 0.3 à 0.9). Il a des options mal écrites (ou incohérentes). Le programme **dialog** mappe ces options obsolètes à celles privilégiées. Cela inclut :

<i>Option</i>	<i>Traitement</i>
--beep-after	ignoré
--guage	mappé à --gauge

Xdialog

C'est une application X plutôt qu'un programme de terminal. En prenant quelques précautions, il est possible d'écrire des scripts utiles qui fonctionnent avec **Xdialog** et **dialog**.

Le programme **dialog** ignore ces options qui sont reconnues par **Xdialog** :

<i>Option</i>	<i>Traitement</i>
--allow-close	ignoré
--auto-placement	ignoré
--fixed-font	ignoré
--icon	ignoré
--keep-colors	ignoré
--no-close	ignoré
--no-cr-wrap	ignoré
--screen-center	ignoré
--separator	mappé à --separate-output
--smooth	ignoré
--under-mouse	ignoré
--wmclass	ignoré

La page de manuel d'**Xdialog** a une section détaillant cette compatibilité avec **dialog**. Quelques différences ne sont pas notées dans cette page. Par exemple, la documentation HTML stipule :

Note : les publications précédentes d'**Xdialog** utilisaient « \n » (nouvelle ligne) comme séparateur de résultats pour le composant

graphique checklist. Cela a été changé en « / » dans la version 1.50 d'Xdialog pour être compatible avec (c)dialog. Dans des anciens scripts utilisant la boîte checklist d'Xdialog, l'option **--separate-output** doit être ajoutée avant l'option **--checklist**.

dialog n'a pas utilisé d'autre séparateur. La différence était probablement due à une confusion concernant certains scripts.

Whiptail

Il existe **whiptail**. Dans la pratique, il est entretenu par Debian (très peu de travail est réalisé par les développeurs amont). Sa documentation (README.whiptail) affiche :

whiptail(1) est un remplaçant léger pour **dialog(1)** pour fournir des boîtes de dialogue pour des scripts d'interpréteur de commandes. Il est construit sur la bibliothèque de fenêtrage newt plutôt que sur la bibliothèque ncurses, permettant d'être plus petit dans des environnements embarqués tels que les installateurs, les disques de récupération, etc. **whiptail** est conçu pour être un remplaçant prêt à l'emploi et compatible avec **dialog**, mais il possède moins de fonctionnalités telles que tailbox, timebox, calendarbox, etc.

En comparant les tailles réelles (Debian testing, 2007/1/10), la taille cumulée pour **whiptail** et les bibliothèques newt, popt et slang est 757 Ko. La taille comparable pour **dialog** (en comptant aussi ncurses) est 520 Ko, contredisant le premier paragraphe.

Le second paragraphe est trompeur puisque **whiptail** ne fonctionne pas pour les options communes de **dialog** telle que la boîte de dialogue gauge. **whiptail** est moins compatible avec **dialog** que le programme dialog 0.4 originel du milieu des années 1990.

La page de manuel de **whiptail** emprunte des fonctionnalités de **dialog**, par exemple, mais curieusement cite les versions de **dialog** jusqu'à la version 0.4 (1994) comme source. C'est-à-dire que sa page de manuel se réfère à des fonctionnalités qui étaient empruntées de versions plus récentes de **dialog**, par exemple :

- **--gauge** (de 0.5) ;
- **--passwordbox** (des modifications de Debian en 1999) ;
- **--default-item** (de **dialog** 22/02/2000) ;
- **--output-fd** (de **dialog** 14/08/2002).

Debian utilise **whiptail** pour la variante officielle de **dialog**.

Le programme **dialog** ignore ou mappe ces options qui sont reconnues par **whiptail** :

<i>Option</i>	<i>Traitement</i>
--cancel-button	mappé à --cancel-label
--fb	ignoré
--fullbutton	ignoré
--no-button	mappé à --no-label
--nocancel	mappé à --no-cancel
--noitem	mappé à --no-items
--notags	mappé à --no-tags
--ok-button	mappé à --ok-label
--scrolltext	mappé à --scrollbar
--topleft	mappé à --begin 0 0
--yes-button	mappé à --yes-label

Il y a des différences de présentation qui ne sont pas corrigées par des options de ligne de commande :

- **dialog** centre la liste dans la fenêtre. **whiptail** typiquement dispose les listes contre la marge gauche ;
- **whiptail** utilise des chevrons « < » et « > » pour le marquage des boutons. **dialog** utilise des crochets ;
- **whiptail** marque les limites des sous-titres avec des barres verticales. **dialog** ne marque pas les limites ;
- **whiptail** essaie de marquer les cellules basse et haute d'une barre de défilement avec des flèches haut et bas. Quand il ne peut le faire, il remplit ces cellules avec la couleur d'arrière-plan et c'est déroutant pour l'utilisateur. **dialog** utilise l'espace en entier de la barre de défilement, par la même fournissant une meilleure résolution.

BOGUES

Peut-être.

EXEMPLES

Les sources de **dialog** contiennent plusieurs exemples sur la manière d'utiliser les différentes options des boîtes de dialogue et à quoi celles-ci ressemblent. Jetez un œil dans le répertoire **samples/** du source.

AUTEURS

Thomas E. Dickey (mises à jour à partir de la version 0.9b).

Kiran Cherupally - composants graphiques mixedform et mixedgauge.

Tobias C. Rittweiler

Valery Reznic - composants graphiques form et progressbox.

Yura Kalinichenko a adapté le composant graphique de jauge ainsi que « pause ».

Il s'agit d'une réécriture (sauf en cas de raison de compatibilité) de la version de **dialog 0.9a** qui liste les auteurs :

- Savio Lam - version 0.3, « dialog » ;
- Stuart Herbert - correctif pour la version 0.4 ;
- Marc Ewing - composant graphique de jauge ;
- Pasquale De Marco (Pako) - version 0.9a, « cdialog ».

TRADUCTION

La traduction française de cette page de manuel a été créée par Gérard Delafond <gerard@delafond.org> et Jean-Paul Guillonnet <guillonnet.jeanpaul@free.fr>

Cette traduction est une documentation libre ; veuillez vous reporter à la [GNU General Public License version 3](#) concernant les conditions de copie et de distribution. Il n'y a aucune RESPONSABILITÉ LÉGALE.

Si vous découvrez un bogue dans la traduction de cette page de manuel, veuillez envoyer un message à debian-l10n-french@lists.debian.org.

22 décembre 2025