

Contents

1 **Module Zenity : Une interface fonctionnelle à l'outil Zenity Dernière version ici:**
<http://besson.qc.to/publis/Zenity/> 1

1 **Module Zenity : Une interface fonctionnelle à l'outil Zenity Dernière version ici:** <http://besson.qc.to/publis/Zenity/>

Author(s): Lilian Besson <lilian.besson[AT]ens-cachan.fr>

`exception ZenityOcaml_Erreur_Annule of int`

L'exception qui est renvoyé si le processus appelé par les fonctions suivantes retourne un code d'erreur UNIX spécifiant une erreur.

`val ecrire_dans_fichier : chaine:string -> adresse:string -> unit`

Ecrit la chaîne `chaine` dans le fichier contenu à l'adresse relative ou absolue `adresse`. Lève une exception si ce fichier est absent.

`val nom_fichier_temporaire : string`

Nom du fichier temporaire où est stocké le résultat de l'appel à Zenity.

`val fairezenity : string -> unit`

Appel `zenity` avec une commande, et fork le résultat dans `nom_fichier_temporaire`

Définition des différentes fonctions utiles du module `ZenityOcaml`

Toutes ces fonctions ont des paramètres uniquement optionnels, qui sont initialisés avec les valeurs par défaut. Chaque fonction a au moins les paramètres communs `~title`, `~window_icon`, `~width`, `~height`, `~expiration`.

`zenity -calendar`

`val calendar :`

`?title:string ->`

`?window_icon:string ->`

`?width:int ->`

`?height:int ->`

`?expiration:int ->`

`?text:string ->`

`?day:int -> ?month:int -> ?year:int -> ?date_format:string -> unit -> string`

Ouvre une demande de sélection de date.

`zenity -entry`

`val entry :`

`?title:string ->`

`?window_icon:string ->`

```
?width:int ->
?height:int ->
?expiration:int ->
?text:string -> ?entry_text:string -> ?hide_text:bool -> unit -> string
```

Ouvre une demande de texte.

```
zenity -password
```

```
val password :
```

```
?title:string ->
?window_icon:string ->
?width:int ->
?height:int -> ?expiration:int -> ?username:string -> unit -> string
```

Ouvre une demande de mot de passe.

```
type add_forms =
```

```
| Add_entry of string
    Ajoute une entrée requête de texte.
| Add_password of string
    Ajoute une entrée requête de mot de passe.
| Add_calendar of string
    Ajoute une entrée requête de date.
| NoAdd
    N'ajoute rien.
```

Type des entrées supplémentaires de la fonction forms (formulaire).

```
val print_add_forms : add_forms -> string
```

Une méthode simple pour afficher des éléments de ce type.

```
zenity -forms
```

```
val forms :
```

```
?title:string ->
?window_icon:string ->
?width:int ->
?height:int ->
?expiration:int ->
?text:string ->
?separator:string ->
?forms_date_format:string -> ?add:add_forms list -> unit -> string
```

Ouvre un formulaire. Voir `add_forms` pour plus de détails.

```
zenity --color-selection
```

```
val color_selection :  
  ?title:string ->  
  ?window_icon:string ->  
  ?width:int ->  
  ?height:int ->  
  ?expiration:int -> ?color:Graphics.color -> unit -> Graphics.color
```

Ouvre une demande de sélection de couleur. Nécessite le module `Graphics`, inclus dans les versions compilées `zenity.cma` et `zenity.cmxa`. Il n'y a normalement rien d'autre à faire pour l'utilisation de `ZenityOcaml`.

```
zenity --scale
```

```
val scale :  
  ?title:string ->  
  ?window_icon:string ->  
  ?width:int ->  
  ?height:int ->  
  ?text:string ->  
  ?value:int ->  
  ?min_value:int ->  
  ?max_value:int ->  
  ?step:int -> ?print_partial:bool -> ?hide_value:bool -> unit -> int
```

Ouvre une demande de valeurs entières.

```
zenity --file-selection
```

```
val file_selection :  
  ?title:string ->  
  ?window_icon:string ->  
  ?width:int ->  
  ?height:int ->  
  ?expiration:int ->  
  ?filename:string ->  
  ?multiple:bool ->  
  ?directory:bool ->  
  ?save:bool ->  
  ?separator:string ->  
  ?confirm_overwrite:bool -> ?file_filter:string -> unit -> string list
```

Ouvre une demande de sélection de fichier dans un explorateur de dossiers.

```
zenity --list
```

```
val list :
```

```

?title:string ->
?window_icon:string ->
?width:int ->
?height:int ->
?expiration:int ->
?text:string ->
?column:string ->
?checklist:bool ->
?radiolist:bool ->
?separator:string ->
?multiple:bool ->
?editable:bool ->
?print_column:int ->
?hide_column:int ->
?hide_header:bool -> ?liste:string list -> unit -> string list

```

Ouvre une demande de sélection dans une liste construite par l'utilisateur (argument liste). Le paramètre separator est inutile. Les paramètres non documentés ne sont pas encore bien supportés (bêta).

zenity -text-info

```

val text_info :
  ?title:string ->
  ?window_icon:string ->
  ?width:int ->
  ?height:int ->
  ?expiration:int ->
  ?filename:string ->
  ?editable:bool ->
  ?display:int ->
  ?html:bool ->
  ?url:string -> ?checkbox:string -> ?font:string -> unit -> string

```

Ouvre une fenêtre informative contenant du texte.

Edit

```

val edit :
  string ->
  ?title:string ->
  ?window_icon:string ->
  ?width:int ->
  ?height:int ->
  ?expiration:int -> ?display:int -> ?font:string -> unit -> string

```

Fournit un éditeur de texte minimal. Le premier paramètre sans nom correspond à l'adresse relative ou absolue du fichier à éditer. Bien sur, les fonctionnalités d'un tel éditeur sont très limitées.

```

zenity --error
val error :
  ?title:string ->
  ?window_icon:string ->
  ?width:int ->
  ?height:int ->
  ?expiration:int -> ?text:string -> ?no_wrap:bool -> unit -> unit
  Ouvre une fenêtre d'erreur. Lève une exception si l'utilisateur n'appuie pas sur le bouton
  'positif'.

zenity --info
val info :
  ?title:string ->
  ?window_icon:string ->
  ?width:int ->
  ?height:int ->
  ?expiration:int -> ?text:string -> ?no_wrap:bool -> unit -> unit
  Ouvre une fenêtre informative. Lève une exception si l'utilisateur n'appuie pas sur le bouton
  'positif'.

zenity --question
val question :
  ?title:string ->
  ?window_icon:string ->
  ?width:int ->
  ?height:int ->
  ?expiration:int ->
  ?text:string ->
  ?ok_label:string -> ?cancel_label:string -> ?no_wrap:bool -> unit -> unit

  Ouvre une demande de réponse à une question. Lève une exception si l'utilisateur n'appuie pas
  sur le bouton 'positif'.

zenity --warning
val warning :
  ?title:string ->
  ?window_icon:string ->
  ?width:int ->
  ?height:int ->
  ?expiration:int -> ?text:string -> ?no_wrap:bool -> unit -> unit
  Ouvre une fenêtre d'avertissement. Lève une exception si l'utilisateur n'appuie pas sur le
  bouton 'positif'.

zenity --notification
val notification :

```

```
?title:string ->
>window_icon:string ->
>width:int ->
>height:int -> ?expiration:int -> ?text:string -> ?listen:bool -> unit -> int
```

Ouvre une fenêtre d'avertissement. Ne fait rien si text est vide.

```
type urgency_level =
| Low
| Normal
| Critical
```

Réprésente les trois niveaux d'urgence utilisées pour la fonction notify.

```
val print_urgency_level : urgency_level -> string
    notify
val notify :
    ?urgency:urgency_level ->
    ?expiration:int ->
    ?icon:string -> ?body:string -> ?summary:string -> unit -> unit
```

Affiche une notification système, dans la zone prévue à cet effet. summary doit être non vide.

```
val help : string -> string
```

Affiche l'aide de zenity concernant le topic argument.