

CLAWS MAIL



Claws Mail

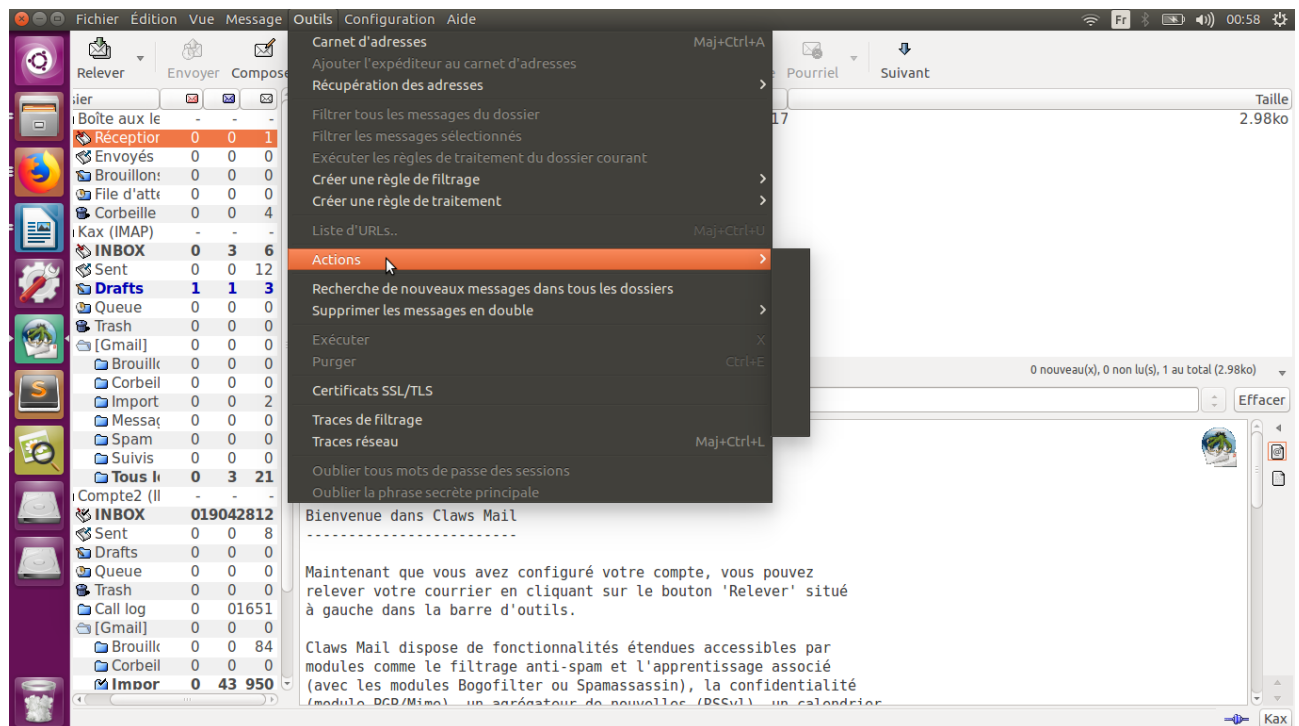
Sommaire

CLAWS MAIL.....	1
Choix du bug/feature.....	2
Arborescence des fichiers concernés.....	4
Rôle des fonctions.....	4
Lien dans le code.....	10
Recherche de solutions.....	11

Choix du bug/feature

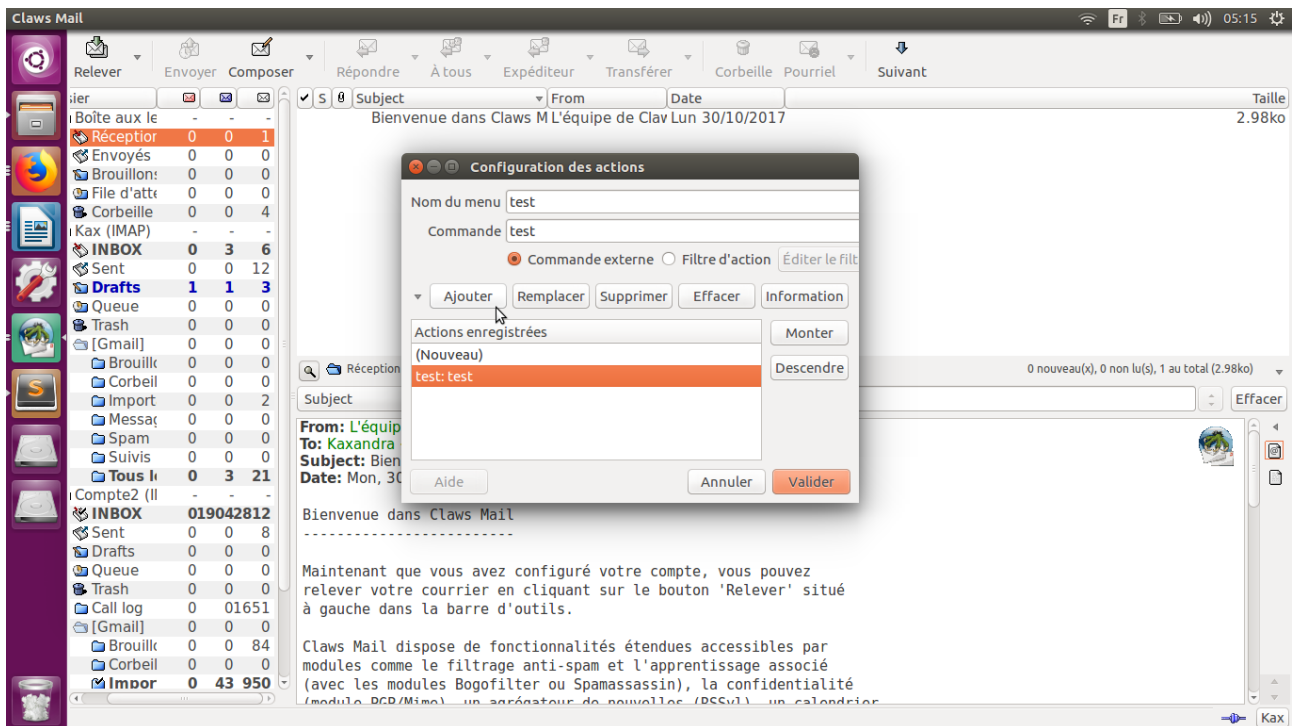
En explorant toutes les fonctionnalités de Claws Mail 3.15.1, qui est la dernière version en date, nous sommes tombés sur une option « Actions » qui, lorsque l'on passe la souris dessus, par défaut, un rectangle noir et vide est affiché. (Outils > Actions)

La fonctionnalité des actions est un moyen pour l'utilisateur d'exécuter des commandes externes depuis Claws Mail. Il est par exemple possible de manipuler physiquement ou graphiquement (vue du message, fenêtre de composition) les messages ou une de leurs parties.



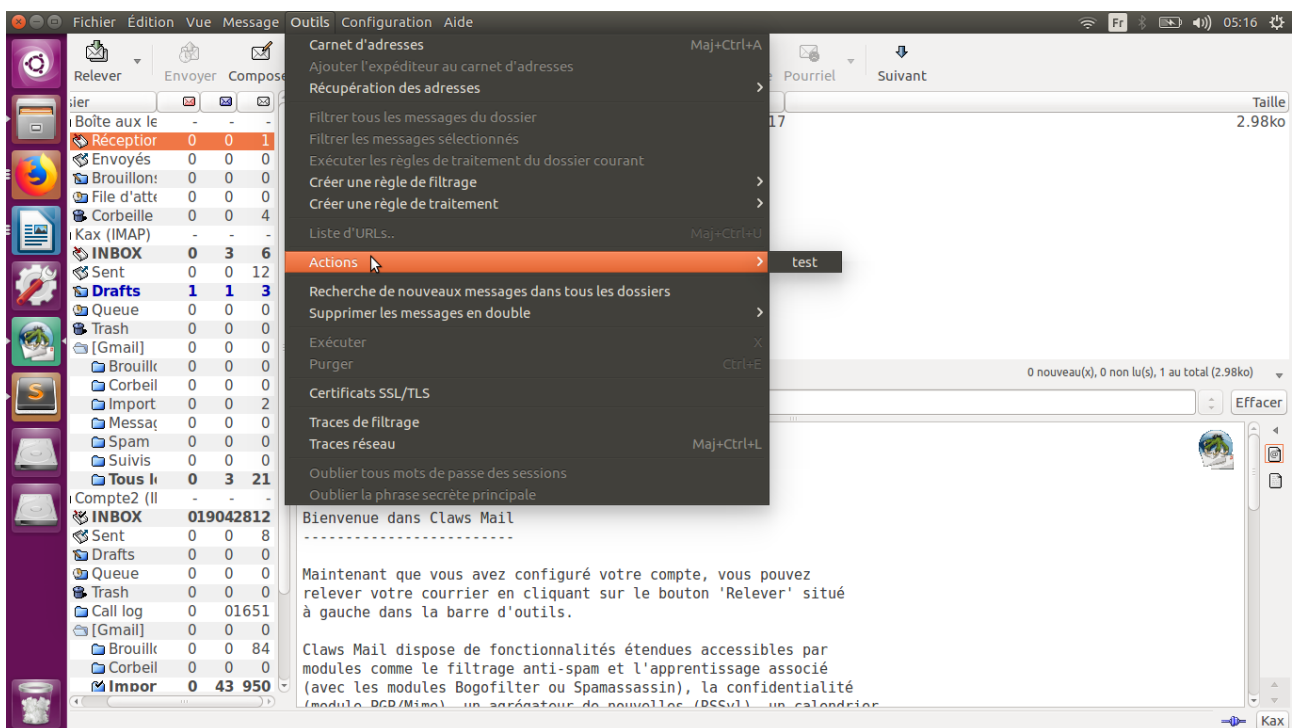
Pour savoir d'où provient ce rectangle vide, nous avons exploré le reste des fonctionnalités, nous sommes allés dans Configuration > Actions.. et nous avons pu voir que l'on pouvait configurer les actions en y rajoutant un nom du menu et une commande et qu'il était possible de choisir entre le fait que ce soit une commande externe ou un filtre d'action.

Nous avons fait un test en rentrant « test » dans nom du menu et « test » dans commande et en laissant commande externe, et nous avons ajouté cette action dans la configuration des actions.



Nous supposons donc par la suite, qu'il s'agirait d'un problème d'affichage par défaut. Lorsque l'on configure une action en y ajoutant un nom et une commande, qui sont obligatoire pour cette fonctionnalité, en retournant dans Outils > Actions, nous voyons que ce « test » est affiché.

Par défaut, si rien n'est configuré dans la configuration des actions, alors il supposerait que dans Outils > Actions, la fonctionnalité « Actions » soit grisée (comme pour la fonctionnalité « Exécuter par exemple) et qu'il n'y ait pas de rectangle noir vide qui s'affiche.



Arborescence des fichiers concernés

1. Où ces fichiers se situent-ils dans l'arborescence ?

```
|— src  
|  |— action.c  
|  |— action.h  
|  |— action.o  
|  |— prefs_actions.c  
|  |— prefs_actions.h  
|  |— prefs_actions.o  
|  |— prefs_filtering_action.c  
|  |— prefs_filtering_action.h  
|  |— prefs_filtering_action.o  
|  |— mainwindow.c  
|  |— mainwindow.h  
|  |— mainwindow.o
```

Rôle des fonctions

2. Quelles fonctions/classes/procédures contiennent-ils ? Quel est leur rôle ?

On a décidé de lister chaque fonction de chaque fichier, celles qui sont en gras et noire sont celles que l'on a pas compris.

Dans le fichier **action.c** se trouvent 35 fonctions. La bibliothèque Glib est requise pour la compilation de ce fichier. C'est une bibliothèque que nous devons installer pour pouvoir aboutir à la fin de l'installation de Claws Mail. GLib est une bibliothèque libre de manipulation de structures de données, d'arbres, de table de hachage et de listes.

ActionType action_get_type : cette fonction permet de savoir quel genre d'action l'utilisateur va choisir.

static gchar *parse_action_cmd : cette fonction permet d'analyser la commande rentrée dans la configuration des actions.

static gboolean parse_append_filename : cette fonction permet d'analyser l'ajout d'un nom de fichier et vérifie donc s'il est bien rentré, sinon renvoie un message d'erreur.

static gboolean parse_append_msgpart : cette fonction permet d'analyser l'ajout d'un message

void actions_execute : cette fonction permet d'exécuter des actions et seront affichées dans la fenêtre principale

void action_update_mainwin_menu : cette fonction permet de rafraichir la fenêtre principal du menu

action_update_msgview_menu : cette fonction permet de rafraichir la vue des messages dans le menu

void action_update_compose_menu : cette fonction permet de rafraichir la composition d'un message dans le menu

static GtkWidget *find_item_in_menu : cette fonction permet de faire une recherche dans le menu. Une interface utilisateur GTK + est construite en imbriquant des widgets dans des widgets. Les widgets de conteneur sont les nœuds internes de l'arbre de widgets résultant: ils contiennent d'autres widgets.

static GtkWidget *create_submenus : cette fonction permet de créer un autre widget contenant des sous menus.

static void action_update_menu : cette fonction permet de rafraichir les actions dans le menu.

static void compose_actions_execute_cb

static void compose_actions_execute : cette fonction exécute les actions dans la fenêtre de composition, s'il y a une erreur, elle renvoie un message d'erreur si l'action sélectionnée ne peut être utilisée dans la fenêtre de composition.

static void mainwin_actions_execute_cb

static void mainwin_actions_execute

static void msgview_actions_execute_cb

static void msgview_actions_execute

static void message_actions_execute : cette fonction exécute les actions dans la fenêtre des messages et les actions filtrées.

static gboolean execute_filtering_actions : cette fonction permet de savoir si une action de filtrage a été demandée, sinon elle renvoie un message d'erreur.

static gboolean execute_actions

static ChildInfo *fork_child

static void kill_children_cb

static gint wait_for_children

static void send_input

static gint delete_io_dialog_cb

static void hide_io_dialog_cb

```
static gint io_dialog_key_pressed_cb
static void childinfo_close_pipes
static void free_children
static void update_io_dialog
static void create_io_dialog
static void catch_status
static void catch_input
static void catch_output
static gchar *get_user_string
```

Ce fichier action.c contient toutes les fonctionnalités des actions à réaliser. Elle est accompagnée de son prototype, action.h et de action.o.

Dans le fichier [prefs-actions.c](#) on trouve 38 fonctions :

void prefs_actions_open : cette fonction est de type void et sert à tester si une action existe, elle va l'ouvrir, sinon elle va la créer.

static void prefs_actions_size_allocate_cb : cette fonction est de type void et sert à enregistrer la taille de l'objet Gtk dans le jeu de données prefs.

static void prefs_actions_create : cette fonction est de type void et permet de créer une action.

static void prefs_actions_reset_dialog : cette fonction est de type void et permet de réinitialiser une action.

void prefs_actions_read_config : cette fonction est de type void et permet la lecture des configurations d'actions.

void prefs_actions_write_config : cette fonction est de type void et permet l'écriture de configuration d'actions.

static void prefs_actions_clear_list : cette fonction est de type void et sert à effacer la liste des actions.

static void prefs_actions_set_dialog : cette fonction est de type void et permet de définir une action.

static void prefs_actions_set_list : cette fonction est de type void et sert à définir la liste des actions.

static gint prefs_actions_clist_set_row : cette fonction est de type gint et permet de mettre une ligne dans la liste des actions.

static void prefs_actions_register_cb : cette fonction est de type void et sert à enregistrer une action.

static void prefs_actions_delete_cb : cette fonction est de type void et sert à supprimer une action.

static void prefs_actions_delete_all_cb : cette fonction est de type void et sert à supprimer toutes les actions configurées.

static void prefs_actions_clear_cb

static void prefs_actions_duplicate_cb

static void prefs_actions_up : cette fonction est de type void et sert à faire monter l'action configurée pour qu'elle soit affichée un rang plus en haut de la liste.

static void prefs_actions_down : cette fonction est de type void et sert à faire descendre l'action configurée pour qu'elle soit affichée un rang en dessous de la liste.

static gint prefs_actions_deleted

static gboolean prefs_actions_key_pressed

static gboolean prefs_actions_search_func_cb

static void prefs_actions_cancel

static void prefs_actions_ok

static void prefs_actions_info_cb

static GtkListStore* prefs_actions_create_data_store

static void prefs_actions_list_view_insert_action

static void prefs_actions_row_selected

static gint prefs_actions_list_btn_pressed

static gboolean prefs_actions_list_popup_menu

static GtkWidget *prefs_actions_list_view_create(void)

static void prefs_actions_create_list_view_columns

static void prefs_actions_select_row

static void prefs_action_filter_radiobtn_cb

static void prefs_action_shell_radiobtn_cb

static void prefs_action_filterbtn_cb

static void prefs_action_define_filter_done

void prefs_actions_rename_path

gint prefs_actions_find_by_name

Le fichier **prefs_filtering_action.c** contient 35 fonctions static, ce qui veut dire qu'elles sont exécutées que dans le fichier et traitent plusieurs filtres qu'on peut ajouter à une action tel que : changer les couleurs, créer une boîte de dialogue, supprimer une action ..etc, parmi ces fonctions on y trouve :

void prefs_filtering_action_open : elle prend comme paramètre la liste d'action qui est action_list avec un pointeur cb qui fait référence à la fonction callback, elle permet d'ouvrir la boîte de dialogue d'action de filtrage avec une liste d'actions ainsi que de mettre à jour le menu des libellés.

static void prefs_filtering_action_size_allocate_cb : permet d'enregistrer la taille de la GTK dans le jeu de données.

static void prefs_filtering_action_create : c'est une fonction générique qui traite deux versions principales, la version 3.0 ainsi que la 2.24 qui permet de créer la boîte de dialogue de correspondance et de les configurer (enregistrer / remplacer / supprimer).

static void prefs_filtering_action_list_view_set_row : cette fonction sert à définir le contenu d'une ligne.

static void prefs_filtering_action_set_dialog : c'est une fonction qui a un seul argument (la liste des action) et elle permet d'initialiser le dialogue avec un ensemble d'actions.

static GSList *prefs_filtering_action_get_list(void) : c'est une fonction générique qui sert à convertir les actions en cours dans la zone d'une liste d'actions utilisée par le système de filtrage et qui retourne cette liste d'actions.

static gint get_account_id_from_list_id : c'est une fonction qui prend la liste des actions comme argument et renvoie l'identifiant de compte.

static gint prefs_filtering_action_get_matching_from_action : elle permet de renvoyer l'ID d'une action de l'analyseur à partir de l'ID d'action interne.

static FilteringAction * prefs_filtering_action_dialog_to_action : c'est une fonction qui permet de renvoyer une action à partir du contenu de la boîte de dialogue, elle prend un booléen (alert) comme paramètre afin de spécifier si les boîtes de dialogue d'alerte doivent être affichées ou pas. Elle traite deux versions la 3.0 ainsi que la 2.24.

static void prefs_filtering_action_register_cb : permet de mieux gérer le signal du bouton d'enregistrement.

static void prefs_filtering_action_delete_cb : permet de mieux gérer le signal du bouton supprimer.

static void prefs_filtering_action_up : permet de mieux gérer le signal du bouton remonter et qui annule tout mouvement si nous sommes à la ligne 0 ou 1.

static void prefs_filtering_action_down : permet de mieux gérer le signal du bouton déplacer vers le bas quand on ne se situe pas à la ligne 0.

static gboolean prefs_filtering_action_key_pressed : elle prend trois arguments, une interface graphique, les événements et la liste des utilisateurs dans la base de données. Elle permet de manipuler les touches du clavier et renvoie une valeur booléenne.

static void prefs_filtering_action_cancel : cette fonction générique permet d'annuler le dialogue de correspondance.

static void prefs_filtering_action_ok : sert à valider les coupleurs actuels.

static gint prefs_filtering_action_deleted : c'est une fonction qui prend aussi trois paramètres qui sont : l'interface graphique, les événements ainsi que les données de l'utilisateur, elle est appelée lors de la fermeture de dialogue et renvoie une valeur booléenne TRUE une fois que le travail est fait.

static void prefs_filtering_action_exec_info : elle prend comme argument widget qui est une interface graphique ainsi qu'un processus père. Cette fonction sert à afficher les informations de l'action Exécuter.

static void prefs_filtering_action_list_view_insert_action : elle prend 4 arguments qui sont la liste des actions, la ligne, l'action à ajouter ainsi qu'une variable booléenne afin de vérifier si l'action à insérer n'existe pas déjà, pour effectuer la vérification la fonction fait appel à une autre fonction qui est `gtk_list_store_set`.

static void prefs_filtering_action_select_addressbook : sert à sélectionner une action de filtrage sur un carnet d'adresses.

static void prefs_filtering_action_enable_widget : cette fonction sert à activer l'action de filtrage.

static void prefs_filtering_action_type_selection_changed : permet de modifier le type d'une action de filtrage.

static void prefs_filtering_action_reset_dialog : c'est une fonction générique qui permet de réinitialiser la boîte de dialogue.

static GtkWidget*prefs_filtering_action_list_view_create(void) : c'est une fonction qui permet de renvoyer la liste d'action créée avec ces filtres.

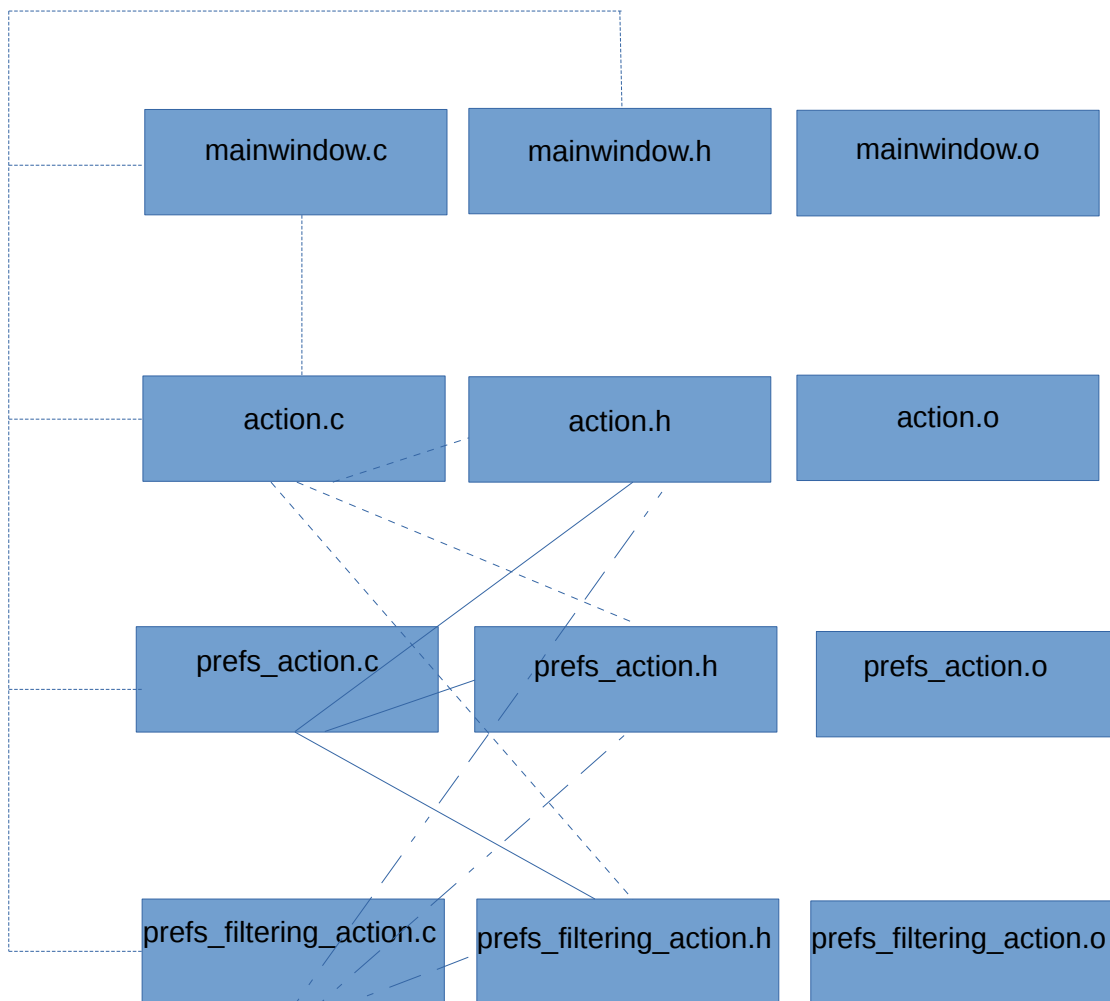
static void prefs_filtering_action_create_list_view_columns : cette fonction sert à créer des colonnes et elle est appelée dans la fonction : `staticGtkWidget*prefs_filtering_action_list_view_create`.

static gboolean prefs_filtering_actions_selected : c'est une fonction récursive qui sert à vérifier si une action est sélectionnée ou pas et renvoie une valeur booléenne à la fin de son exécution.

Lien dans le code

3. Comment sont-elles reliées au reste du code : qui les utilise ? qu'utilisent-elles ?

Ces fonctions sont reliées au reste du code présent dans le fichier src par des séries d' « include » dans chaque fichier.



Recherche de solutions

Ce n'est pas tout, l'aspect graphique à un rôle dans notre bug, c'est à dire qu'il y a bien un fichier contenant des fonctions liées à l'interface, au visuel, et ce fichier en question, c'est `mainwindow.c` (il y a un `.h` également, son header).

`Mainwindow.c` est le fichier chargé de structurer notre interface de consultation de mail, (l'interface ou sur laquelle on va passer le plus de temps), on y retrouve les outils, ainsi que toutes les fonctionnalités que l'on découvre en ouvrant Claws Mail.

C'est l'un des fichiers à modifier pour pouvoir « griser » l'outil « actions » par défaut.

Par sécurité, nous n'avons pas modifié ces fichiers car il se peut que cela ait un impact sur d'autres fonctionnalités importantes de Claws Mail et que l'on ne saurait corriger malheureusement.