


# Mcabber : client XMPP console

- Objet : Prise en main du client de messagerie Mcabber
- Niveau requis :  
[débutant, avisé](#)
- Commentaires : *Utilisation du protocole XMPP en console.*
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande, tout commence là !](#) 😊
- Suivi :  
[à-tester](#)
  - Création par  [Thuban](#) le 07/06/2013
  - Testé par <...> le <...>
- Commentaires sur le forum : [ici](#)<sup>1)</sup>

## Présentation

[Mcabber](#) est un client de messagerie instantanée pour le protocole [XMPP](#), que vous connaissez peut-être mieux sous le terme [Jabber](#).

Il se démarque des autres par son utilisation en console, ce qui le rend d'une extrême légèreté, sans pour autant lui retirer ses fonctionnalités.

Si vous utilisez principalement la messagerie instantanée pour envoyer/recevoir des messages, et que vous n'avez pas besoin des fonctions vidéo/audio de XMPP, alors mcabber sera une solution très efficace pour vous!

## Installation

Accrochez vos ceintures, pour installer mcabber, il faut lancer cette commande :

```
apt-get update && apt-get install mcabber
```

## Utilisation

La fenêtre se présente divisée en deux : à gauche la liste des contacts, que l'on appellera le roster, à droite la fenêtre de discussion.

```
[status]
--- Groupe 1
[a] contact@jab
[o] contact2@ja
[_] contact3@ja
```

Informations diverses... salon, statut,...		
Là où vous écrivez		

Dans le roster, on peut connaître le statut des contacts d'après le symbole entre [ ]. Voici leurs signification :

<b>STATUTS</b>	<b>ACTION</b>
o	En ligne (online)
f	Prêt à discuter (free)
d	Ne pas déranger (do not disturb)
n	Indisponible (not available)
a	Absent (away)
_	Déconnecté
?	Status inconnu
x	Salon de discussion où vous ne participez pas
C	Salon de discussion où vous participez

Il existe un contact particulier : [status]. Ce dernier vous donne toutes les indications de mcabber, lorsque vous demandez de l'aide avec /help par exemple...

Pour sélectionner un contact à qui parler, vous pouvez vous déplacer dans le roster avec les touches **PgPrec** ou **⌕** ou **Page↑** et **PgSuiv** ou **⌕** ou **Page↓**

Pour voir les anciens messages, on peut se déplacer dans l'historique avec **Ctrl+P** et **Ctrl+N** Pour écrire à un contact, tapez simplement votre message puis **↵ Entrée**

Enfin, sachez que mcabber donne accès à de nombreuses fonctionnalités via des commandes commençant par un /. Rassurez-vous, pas besoin de toutes les retenir! Il suffira de taper /help pour avoir une liste des commandes disponibles. Ensuite, taper /help commande vous donnera des explications sur l'utilisation de cette commande. Par exemple, si on tape /help move, on obtient dans le contact [status]

/MOVE [groupname]

06-07 08:56:01 Déplace le contact sélectionné vers le groupe spécifié. Si aucun groupe n'est donné, le contact est déplacé vers le groupe par défaut. Si le groupe "groupname" n'existe pas, il est créé.

06-07 08:56:01 Astuce : si le mode discussion (chatmode) est activé, vous pouvez utiliser "/roster alternate" pour vous positionner sur le contact que vous venez de déplacer.

Voyons quelques-unes de ces commandes les plus utiles.

Astuce : Tapez `Tab ↵` et la commande sera auto-complétée.

## Commande /add

Cette commande permet d'ajouter un contact.

```
/add contact@jabber.ru Surnom
```

## Commande /authorization

Cette commande vous permet d'autoriser un contact à vous voir ou non, demander son statut...

Sélectionnez un contact, puis selon ce que vous voulez :

```
/authorization allow  
/authorization cancel  
/authorization request
```

Plus de détails en lançant cette commande `/help authorization`

## Commande /buffer

Cette commande vous permet de manipuler l'historique du texte tapé. Par exemple pour chercher du texte :

```
/buffer search_backward textecherché
```

Pour enregistrer dans un fichier l'historique :

```
/buffer save nom_du_fichier
```

Toujours plus de détails avec `/help buffer`

## Commande /del

Supprimer un contact. Et oui, c'est brutal aussi mcabber! :)

## Commande /move

Cette commande permet de déplacer le contact sélectionné dans un groupe. Un groupe inexistant sera alors créé.

```
/move Salons
```

## Commande /room

Cette commande permet de gérer les salons. Elle contient de très nombreuses fonctionnalités, je vous conseille donc de taper `\help room`. Voici quelques exemples :

Rejoindre un salon (pseudo et motdepasse facultatifs):

```
/room join salon@jabberfr.org pseudo motdepasse
```

Ajouter aux marques-page et rejoindre automatiquement le salon sélectionné :

```
/room bookmark +autojoin
```

## Configuration

La configuration de mcabber se réalise dans le fichier `~/.mcabberrc`. Copions le fichier exemple fourni avec debian :

```
gunzip -c /usr/share/doc/mcabber/examples/mcabberrc.example.gz >
~/.mcabberrc
```

On se retrouve avec un fichier contenant la configuration par défaut. Remarquez qu'il est très bien commenté. Nous détaillerons ici les options essentielles, mais vous pouvez peaufiner la configuration de vous-même.

## Compte et mot de passe.

Pour définir son compte et mot de passe jabber :

```
# Please provide your Jabber (XMPP) identifier:
set jid = moua@jabber.fr
set password = supermegamotdepasse
```

## Multi-compte

J'ai plusieurs compte moi! Comment je fais?

Pas de panique! Il va juste falloir créer un fichier de configuration par compte. Par exemple : `~/.mcabberrcjabberfr` et `~/.mcabberrcgmail`.

Dans chacun d'entre eux vous préciser vos identifiants/mots de passe relatifs au compte. Et maintenant, pour utiliser le compte gmail, vous lancez la commande :

```
mcabber -f ~/.mcabberrc_gmail
```

L'option -f permet simplement de préciser à mcabber quel fichier de configuration utiliser.

## Pseudonyme

Pour avoir son bon pseudonyme dans les salons :

```
# the jid variable can be used.  
set nickname = "MonPseudoQuiDechire"
```

## Interface dans la langue de Victor Hugo

```
set lang = fr
```

## Correction orthographique et encodage

Pour n'avoir aucun soucis d'affichage des caractères, on préférera l'UTF-8 :

```
set spell_enable = 1  
set spell_lang = fr_FR  
set spell_encoding = UTF-8
```

## Qui afficher dans le roster?

On peut choisir le statut des contacts à afficher dans le roster. Par exemple, je ne veux afficher que les contacts en lignes o, et disponibles pour discuter f :

```
set roster_display_filter = of
```

Rappelons la signification de ces lettres :

<b>STATUTS</b>	<b>ACTION</b>
o	En ligne (online)
f	Prêt à discuter (free)
d	Ne pas déranger (do not disturb)
n	Indisponible (not available)
a	Absent (away)
_	Déconnecté
?	Status inconnu
x	Salon de discussion où vous ne participez pas
C	Salon de discussion où vous participez

## Enregistrement des discussions

```
set logging = 1
set load_logs = 1
set logging_dir = ~/.mcabber/histo/
set logging_ignore_status = 1
```

## Modules

Mcabber peut être étendu par des [modules](#). Pour préciser le répertoire où ils sont enregistrés :

```
set modules_dir = /usr/lib/mcabber/
```

Pour charger un module:

```
module load nomdumodule
```

## Temps avant d'être considéré "parti"

Changez le temps avant que votre statut soit automatiquement "away"

```
set autoaway = 180
```

## Changer les messages de statut

```
set message_avail      = Disponible
set message_free       = Prêt à discuter
set message_dnd        = Occupé à autre chose
set message_notavail   = Indisponible
set message_away       = Parti
```

## Auto-complétion des pseudos

Vous pouvez déjà profiter de l'auto-complétion avec la touche `Tab ↵`. Mais mcabber vous donne la possibilité d'avoir l'auto-complétion pour les pseudos des personnes dans un salon!

Décommentez cette ligne dans le `.mcabberrc` :

```
set muc_completion_suffix = ": "
```

Maintenant, pour profiter de cette fonction, tapez "smo:" puis `Tab ↵` et mcabber remplacera par "smolski" . (sous réserve qu'un dénommé smolski soit présent dans le salon) :)

## Fond transparent

Pour que le fond de mcabber soit transparent, précisez ceci dans les couleurs :

```
set color_background = default
```

## Définir ses propres commandes

Vous pouvez vous créer vos propres raccourcis en indiquant vos commandes. Par exemple :

```
alias me = say /me
```

## Raccourcis clavier

Définissez vos propres raccourcis claviers avec `bind keycode action`. Pour connaître le code d'une touche ou combinaison de touche, tapez la simplement dans mcabber. Si elle n'est attribuée à aucun raccourci, alors le code correspondant apparaîtra dans la barre de status. Quelques exemples :

```
# F5 (269) affiche/cache les contacts déconnectés
bind 269 = roster toggle_offline
# F12 (276) affiche/cache le roster
bind 276 = roster toggle
# alt-shift-k Monte dans le roster
bind M75 = roster up
# alt-shift-j descend dans le roster
bind M74 = roster down
```

## Gestion des évènements

Lorsqu'un évènement survient (connexion d'un contact, nouveau message), mcabber peut lancer un script quelconque. Pour cela, on ajoute cette option :

```
set events_command = ~/.mcabber/eventcmd
```

C'est donc le script `eventcmd` placé dans `~/.mcabber` qui sera exécuté. Un exemple de ce script est situé ici : `/usr/share/doc/mcabber/examples/contrib/events/eventcmd`.

Cette solution est très pratique pour avoir des notifications par exemple.

## Raccourcis d'urls

Le principe est le même qu'au paragraphe précédent : lorsqu'un message est reçu, on lance un script. Ce dernier est situé ici :

`/usr/share/doc/mcabber/examples/contrib/events/eventcmd.pastebin`.

## Lancer une commande à un moment donné (hooks)

Le système de “hooks” permet de lancer une action lorsqu'un évènement survient.

Pour lancer une commande lorsque vous vous connectez, par exemple mettre votre statut à “ne pas déranger”:

```
set hook-post-connect = status dnd
```

Et lorsque vous vous déconnectez, pour dire au-revoir à foo@bar :

```
set hook-pre-disconnect = say_to foo@bar Goodbye!
```

## Gérer mcabber à distance

Sans être dans la fenêtre de mcabber, vous pouvez quand même le contrôler. Cela peut être utile sur un serveur éventuellement. Utilisez alors le système de “fifo” :

```
set fifo_name = ~/.mcabber/mcabber.fifo
set fifo_hide_commands = 0
set fifo_ignore = 0
```

Et envoyez ensuite vos commandes dans le fichier fifo `echo commande » ~/.mcabber/mcabber.fifo`

Par exemple, pour changer votre status en “Occupé : En train de réviser”:

```
echo "status dnd En train de réviser" >> ~/.mcabber/mcabber.fifo
```

## IRC

Non, mcabber n'est pas un client IRC. Cependant, certains serveurs jabbers proposent des passerelles vers le protocole IRC. Si votre adresse jabber le permet, alors vous pouvez rejoindre un salon de cette façon :

```
/room join #salon%irc.freenode.net@irc.jabberfr.org
```

Décomposons cette ligne : - /room join : commande mcabber pour rejoindre un salon - #salon : nom du salon IRC - %irc.freenode.net : le serveur IRC est sur irc.freenode.net. Il est important d'utiliser un % (spécifique à jabber) - @irc.jabberfr.org : C'est la passerelle IRC de votre compte jabber. L'exemple utilise celui de jabberfr:

Des informations sur le transport sont [disponibles ici](#)

Oui mais problème, sur IRC, on utilise des commandes du type “/commande”. Comment faire pour que mcabber ne croit pas que c'est une commande pour lui?



Tout bêtement, il faut ajouter devant /say. Par exemple :

```
/say /me
```

Ou encore

```
/say /nick monnouveaupseudo
```

Ajouter des alias dans le fichier de configuration de mcabber s'avère alors très utile.

## Identification sur IRC

Vous pouvez rajouter un alias dans ~/.mcabberrc :

```
alias ircidentify = say_to NickServ%irc.freenode.net@irc.jabberfr.org  
identify VotrePseudo motDePasse
```

Et ensuite, n'avoir qu'à lancer la commande

```
/ircidentify
```

## Auto-login sur IRC

Vous devez peut-être authentifier votre pseudo sur IRC. Il est confortable de le faire automatiquement. Pour cela, une méthode possible avec mcabber est d'utiliser le script d'évènements, avec le fifo de mcabber.

Donc assurez-vous de copier /usr/share/doc/mcabber/examples/contrib/events/eventcmd dans votre dossier ~/.mcabber.

Ensuite, modifiez ~/.mcabberrc pour activer le fifo et le script d'évènements :

```
set events_command = ~/.mcabber/eventcmd  
set fifo_name = ~/.mcabber/mcabber.fifo
```

Dans le script eventcmd, en dessous la section concernant les STATUS, nous allons ajouter quelques lignes, pour vérifier lorsqu'on est connecté à la passerelle IRC, puis dans ce cas envoyer la commande d'authentification via le fifo de mcabber :

```
elif [ $event = "STATUS" ]; then  
    # Buddy $arg2 status is $arg1 (_, O, I, F, D, N, A)  
    #auto-identify on irc  
    if [ $arg2 = "irc.jabberfr.org" ]; then  
        if [ "$arg1" = "O" ]; then  
            sleep 16 # pour être sûr de la connexion.  
            echo "say_to NickServ%irc.freenode.net@irc.jabberfr.org identify  
VotrePseudo motDePasse" >> ~/.mcabber/mcabber.fifo  
        fi  
    fi  
fi
```

fi  
fi

## Pour aller plus loin...

- `man mcabber`
- [Site officiel](#)
- [Guide officiel](#)
- [Dernière version de ce document](#)

<sup>1)</sup>

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

<http://debian-facile.org/doc:reseau:mcabber>

Last update: **08/09/2015 17:50**

