WeeChat : client IRC console

- Objet : **WeeChat**, client IRC en mode texte.
- Niveau requis :
  débutant, avisé
- Commentaires : *Client IRC moderne en mode texte (curses).*
- Débutant, à savoir : Utiliser GNU/Linux en ligne de commande, tout commence là !
  - Le terminal ou console : Utilisation du terminal.
- Suivi :
  - Création par MaTTuX_ le 10/02/2009
  - Mis à jour par MicP le 06/02/2014 ; SuShY le 21/07/2020 ; Nietzsche le 25/11/2020 ; arpinux le 31/05/2021
  - Testé par Nietzsche le 25/11/2020
- Commentaires sur le forum : ici[1)]

Introduction

WeeChat (Wee Enhanced Environment for Chat) est un client Internet Relay Chat gratuit et à source ouverte, conçu pour être léger et rapide. Il est publié sous les termes de la licence publique générale GNU 3 et a été développé depuis 2003. C'est un client de discussion en mode texte utilisant le protocole IRC. Il est léger et rapide, Tout peut y être contrôlé au clavier. Il est extensible par des extensions (plugins) et des scripts.

Un émulateur de terminal recommandé pour X (mais pas obligatoire) est rxvt-unicode (il a un bon support UTF-8, et aucun problème avec les touches par défaut).
Il supporte les scripts écrit en ruby, perl, python et lua grâce à un système de plugins.

- Voir le tuto OFTC pour compléter les instructions

Installation

En console, il suffit de taper :

```
apt-get update && apt-get install weechat
```

fonctionnalités :
- 256 couleurs
- proxy IRC (extension relay)
- redirection des commandes IRC
- l'extension rmodifier
- les expressions rationnelles pour les highlights
- support de la couleur dans l'heure du tampon de discussion
- option IRC pour forcer la couleur de certains pseudos
- partage de la zone de saisie entre les tampons.

Configuration

Depuis la version 3.0, la configuration se fait dans plusieurs fichiers. Aussi, pour simplifier la configuration de ce logiciel, je vous propose de le faire directement dans la fenêtre weechat.

Démarrer et quitter weechat

Dans un terminal sous user, saisissez :

```
weechat-curses
```

Pour quitter weechat :

```
/quit
```

Si votre weechat tourne dans un screen sur un serveur vous pouvez ouvrir ce dernier directement en créant un raccourcis executant cette commande:

```
terminator -e ssh user@ipserveur ou toto.com -t screen -x
```

Ici j'ai mis terminator car c'est l'émulateur de terminal que j'utilise. Cette commande ouvre terminator et y lance la commande de ce fait terminator vous demande directement votre passephrase si vous en avez une ou votre passwd et vous vous retrouverez dans votre weechat.

Pour commencer il faut ajouter le serveur auquel on souhaite se connecter : (par exemple le serveur irc.oftc.net)

```
/server add oftc irc.oftc.net
```

On définit ensuite les pseudos que l'on utilisera :

```
/set irc.server.oftc.nicks "pseudo1,pseudo2,pseudo3,pseudo4,pseudo5"
```

Pour modifier le nom d’utilisateur et le nom réel :

```
/set irc.server.oftc.username "Mon nom d'utilisateur"
/set irc.server.oftc.realname "Mon nom réel"
```

Pour activer la connexion automatique à ce serveur :

```
/set irc.server.oftc.autoconnect on
```

Pour activer l'autorejoin automatique à ce serveur :

```
/set irc.server.oftc.autorejoin on
```

Pour l'identification, on va tricher : on lance une commande après la connexion au serveur :

```
/set irc.server.oftc.command "/msg nickserv identify xxxxxx"
```

Pour configurer les canaux à rejoindre :

```
/set irc.server.oftc.autojoin "#debian-facile,#debian-france"
```

Si vous avez installé **avant** un serveur oftc :
Pour supprimer la valeur d'une option du serveur et utiliser sa valeur par défaut à la place, par exemple pour utiliser les pseudos par défaut (irc.server_default.nicks) :

```
/set irc.server.oftc.nicks null
```

Voilà, on a fait l'essentiel. **N'oubliez pas de taper :**

```
/save
```

afin de ne pas perdre ce que vous venez de faire !

S'il vous vient le désir de modifier cette configuration, il suffit de reprendre la commande où ce situe ces changements à faire et de les exécuter !

Commandes utiles :

Pour se connecter à oftc :

```
/connect oftc
```

Pour joindre un canal :

```
/join #channel
```

Pour quitter un chan en laissant un message :

```
/part [message de fin]
```

Pour quitter le chan et fermer le tampon en laissant un message de fin :

```
/close [message de fin]
```

Pour se marquer absent et éventuellement ajouter un petit commentaire avec :

```
/away parti aux toilettes
```

Pour se démarquer absent, c'est la même mais sans commentaire :

```
/away
```

Pour changer la couleur de son pseudo dans weechat :

```
/set weechat.color.chat_nick_self _blue
```

Démarrer en haut des buffers :

Si lorsque `weechat` démarre on se retrouve tout en bas de la liste des serveurs et salons mais qu'il est préférable que ce soit l'opposé, c'est à dire un démarrage au top de la liste, faire la manipulation suivante :

```
/set irc.look.buffer_switch_autojoin off
```

Sans oublier la sauvegarde :

```
/save
```

Split des fenêtres

Une fonctionnalité très intéressante de weechat est le split des fenêtres. Qu'est-ce que c'est le split des fenêtres ? C'est ça :

Cela vous permettra de suivre plusieurs discussions en même temps :P

Alors :

Pour séparer l'écran horizontalement en deux parties égales :

```
 /window splith
```

Pour séparer l'écran verticalement en deux parties égales :

```
 /window splitv
```

Pour découper verticalement l'écran en une petite fenêtre (1/3 de la largeur) et une large (2/3), utilisez la commande :

```
 /window splitv 33
```

C'est bien beau tout ça, mais comment on navigue entre les fenêtres ?
Deux solutions :

1. En utilisant les **raccourcis claviers** (cf plus bas)
2. En utilisant les commandes ci-dessous :

```
 /window right
 /window left
```

```
  /window up
  /window down
  /window list
```

Et si je veux dégager un des splits ?
Tout simplement :

```
/window merge
```

Pour sauvegarder l'affichage des fenêtres (hors querry)

```
/layout store
/save
```

Raccourcis clavier

WeeChat utilise un certain nombre de touches par défaut. Toutes ces touches sont dans la documentation, mais vous devriez connaître au moins les touches vitales :

RACCOURCIS CLAVIER	***ACTION***
Alt + ← / → (ou F5 / F6)	aller au chan(tampon) précédent/suivant
F7 / F8	aller à la fenêtre précédente/suivante (quand l'écran est divisé)
F9 / F10	faire défiler la barre de titre
F11 / F12	faire défiler la liste des pseudos
Tab	compléter le texte de la barre d'entrée, comme avec votre shell
Page haut / Page bas	faire défiler le texte verticalement dans le tampon courant
← / →	faire défiler de haut en bas le texte dans le tampon courant
Alt + A	sauter au tampon avec de l'activité (dans la "hotlist")
Ctrl + N	sauter au tampon précédent)
Ctrl + P	sauter au tampon suivant)

Selon votre clavier et/ou vos besoins, vous pouvez associer n'importe quelle touche à une commande avec la commande /key. Une touche utile est meta-k (Alt + K) pour trouver le code des touches.

Exemple pour associer meta-y (Alt + Y) à la commande /buffer close :

```
/key bind (pressez meta-k) (pressez meta-y) /buffer close
```

Vous pouvez créer vous-même vos raccourcis :

```
/key bind meta-y /buffer close
```

Pour désactiver le raccourci :

```
/key unbind meta-y
```

WEECHAT 0.3.0 - Scripts

Vous pouvez récupérer des scripts pour weechat (regardez bien les versions weechat indiquées) ici :

- Scripts disponibles

Il vous suffira de les copier dans leurs repertoires respectifs, *python* avec *python*, *perl* avec *perl*, *ruby* avec *ruby*... dans les répertoires **autoload**

Bip sur haut-parleur sur highlight/message privé.

Script: beep.pl

```
nano ~/.weechat/perl/autoload/beep.pl
```

- Télécharger script: beep.pl

Réduire la longueur des URLs en utilisant isgd ou tinyurl.

Je vous propose à présent l'installation d'un script qui réduit la longueur des URLs.

Script: shortenurl.py

Fermer weechat.

Copier le script dans :

```
nano ~/.weechat/python/autoload/shortenurl.py
```

voir nano

- Télécharger script: shortenurl.py

Affichage de la barre d'url pour clic ou sélection facile.

Ce script affiche une barre d'URLs longue.

Nota : Je ne suis pas certain que ce script serait utile.

Script: urlbar.py

```
nano ~/.weechat/python/autoload/urlbar.py
```

- Télécharger script: urlbar.py

Copiez **par exemple:**

~/.weechat/python/autoload/urlbar.py

```
# -*- coding: utf-8 -*-
#
# Copyright (c) 2009 by FlashCode <flashcode@flashtux.org>
# Copyright (c) 2009 by xt <xt@bash.no>
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program.  If not, see <http://www.gnu.org/licenses/>.
#

#
# Bar with URLs (easy click on long URLs)
# (this script requires WeeChat 0.3.0 or newer)
#
# History:
# 2010-12-20, xt <xt@bash.no>
#     version 10: use API for nick color, strip nick prefix
# 2009-12-17, FlashCode <flashcode@flashtux.org>
#     version 0.9: fix option name "show_index" (spaces removed)
# 2009-12-12, FlashCode <flashcode@flashtux.org>
#     version 0.8: update WeeChat site
# 2009-11-05, xt <xt@bash.no>
#     version 0.7: config option to turn off index
# 2009-10-20, xt <xt@bash.no>
#     version 0.6: removed priority on the bar
# 2009-07-01, xt <xt@bash.no>
#     version 0.5: changed script command to /urlbar, comma separated
ignore list
# 2009-05-22, xt <xt@bash.no>
#     version 0.4: added configurable showing of buffer name, nick and
time
```

```
# 2009-05-21, xt <xt@bash.no>
#     version 0.3: bug fixes, add ignore feature from sleo
# 2009-05-19, xt <xt@bash.no>
#     version 0.2-dev: fixes
# 2009-05-04, FlashCode <flashcode@flashtux.org>
#     version 0.1-dev: dev snapshot
#

SCRIPT_NAME    = "urlbar"
SCRIPT_AUTHOR  = "FlashCode <flashcode@flashtux.org>"
SCRIPT_VERSION = "10"
SCRIPT_LICENSE = "GPL3"
SCRIPT_DESC    = "Bar with URLs. For easy clicking or selecting."
SCRIPT_COMMAND = "urlbar"

settings = {
    "visible_amount"        : '5',     # Amount of URLS visible in
urlbar at any given time
    "visible_seconds"       : '5',     # Amount of seconds URLbar is
visible
    "use_popup"             : 'on',    # Pop up automatically
    "remember_amount"       : '25',    # Max amout of URLs to keep in
RAM
    "ignore"                : 'grep',  # List of buffers to ignore.
(comma separated)
    "show_timestamp"        : 'on',    # Show timestamp in list
    "show_nick"             : 'on',    # Show nick in list
    "show_buffername"       : 'on',    # Show buffer name in list
    "show_index"            : 'on',    # Show url index in list
    "time_format"           : '%H:%M', # Time format
}

import_ok = True
try:
    import weechat
except ImportError:
    print "This script must be run under WeeChat."
    print "Get WeeChat now at: http://www.weechat.org/"
    import_ok = False

import re
from time import strftime, localtime
octet = r'(?:2(?:[0-4]\d|5[0-5])|1\d\d|\d{1,2})'
ipAddr = r'%s(?:\.%s){3}' % (octet, octet)
# Base domain regex off RFC 1034 and 1738
label = r'[0-9a-z][-0-9a-z]*[0-9a-z]?'
domain = r'%s(?:\.%s)*\.[a-z][-0-9a-z]*[a-z]?' % (label, label)
urlRe = re.compile(r'(\w+://(?:%s|%s)(?::\d+)?(?:/[^\])>\s]*)?)' %
(domain, ipAddr), re.I)
```

```
# list of URL-objects
urls = []

# Display ALL, a toggle
DISPLAY_ALL = False



def urlbar_item_cb(data, item, window):
    ''' Callback that prints the lines in the urlbar '''
    global DISPLAY_ALL, urls
    try:
        visible_amount =
int(weechat.config_get_plugin('visible_amount'))
    except ValueError:
        weechat.prnt('', 'Invalid value for visible_amount setting.')

    if not urls:
        return 'Empty URL list'

    if DISPLAY_ALL:
        DISPLAY_ALL = False
        printlist = urls
    else:
        printlist = urls[-visible_amount:]

    result = ''
    for index, url in enumerate(printlist):
        if weechat.config_get_plugin('show_index') == 'on':
            index = index+1
            result += '%s%2d%s %s \r' %\
                (weechat.color("yellow"), index,
weechat.color("bar_fg"), url)
        else:
            result += '%s%s \r' %(weechat.color('bar_fg'), url)
    return result


def get_buffer_name(bufferp, long=False):
    if not weechat.buffer_get_string(bufferp, "short_name") or long:
        bufferd = weechat.buffer_get_string(bufferp, "name")
    else:
        bufferd = weechat.buffer_get_string(bufferp, "short_name")
    return bufferd

class URL(object):
    ''' URL class that holds the urls in the URL list '''

    def __init__(self, url, buffername, timestamp, nick):
        self.url = url
        self.buffername = buffername
```

```
        self.time = strftime(
                weechat.config_get_plugin('time_format'),
                localtime(int(timestamp)))
        self.time = self.time.replace(':', '%s:%s' %
                (weechat.color(weechat.config_string(
weechat.config_get('weechat.color.chat_time_delimiters'))),
                weechat.color('reset')))
        self.nick = irc_nick_find_color(nick.strip('%&@+'))

    def __str__(self):
        # Format options
        time, buffername, nick = '', '', ''
        if weechat.config_get_plugin('show_timestamp') == 'on':
            time = self.time + ' '
        if weechat.config_get_plugin('show_buffername') == 'on':
            buffername = self.buffername + ' '
        if weechat.config_get_plugin('show_nick') == 'on':
            nick = self.nick + ' '

        return '%s%s%s%s' % (time, nick, buffername, self.url)

    def __cmp__(this, other):
        if this.url == other.url:
            return 0
        return 1

def urlbar_print_cb(data, buffer, time, tags, displayed, highlight,
prefix, message):


    buffer_name = get_buffer_name(buffer, long=True)
    # Skip ignored buffers
    for ignored_buffer in
weechat.config_get_plugin('ignore').split(','):
        if ignored_buffer.lower() == buffer_name.lower():
            return weechat.WEECHAT_RC_OK

    # Clean list of URLs
    for i in range(len(urls) -
int(weechat.config_get_plugin('remember_amount'))):
        # Delete the oldest
        urls.pop(0)

    for url in urlRe.findall(message):
        urlobject = URL(url, get_buffer_name(buffer), time, prefix)
        # Do not add duplicate URLs
        if urlobject in urls:
            continue
        urls.append(urlobject)
        if weechat.config_get_plugin('use_popup') == 'on':
            weechat.command("", "/bar show urlbar")
```

```
            # auto hide bar after delay
            try:
                weechat.command('', '/wait %s /bar hide urlbar' %
int(weechat.config_get_plugin('visible_seconds')))
            except ValueError:
                weechat.prnt('', 'Invalid visible_seconds')

        weechat.bar_item_update("urlbar_urls")

    return weechat.WEECHAT_RC_OK


def urlbar_cmd(data, buffer, args):
    """ Callback for /url command. """
    global urls, DISPLAY_ALL

    if args == "list":
        if urls:
            DISPLAY_ALL = True
            weechat.command("", '/bar show urlbar')
            weechat.bar_item_update("urlbar_urls")
        else:
            weechat.prnt('', 'URL list empty.')
    if args == "show":
        weechat.command('', '/bar show urlbar')
    elif args == 'hide':
        weechat.command("", "/bar hide urlbar")
    elif args == 'toggle':
        weechat.command("", "/bar toggle urlbar")
    elif args == 'clear':
        urls = []
    else:
        weechat.command("", "/help %s" % SCRIPT_COMMAND)

    return weechat.WEECHAT_RC_OK

def urlbar_completion_urls_cb(data, completion_item, buffer,
completion):
    """ Complete with URLS, for command '/url'. """
    for url in urls:
        weechat.hook_completion_list_add(completion, url.url,
                                         0,
weechat.WEECHAT_LIST_POS_SORT)
    return weechat.WEECHAT_RC_OK

def irc_nick_find_color(nick):

    color = weechat.info_get('irc_nick_color', nick)
    if not color:
        # probably we're in WeeChat 0.3.0
        color %=
```

```
weechat.config_integer(weechat.config_get("weechat.look.color_nicks_number"))
        color = weechat.config_get('weechat.color.chat_nick_color%02d'%(color+1))
        color = w.color(weechat.config_string(color))
    return '%s%s%s' %(color, nick, weechat.color('reset'))


if __name__ == "__main__" and import_ok:
    if weechat.register(SCRIPT_NAME, SCRIPT_AUTHOR, SCRIPT_VERSION, SCRIPT_LICENSE,
                        SCRIPT_DESC, "", ""):
        # Set default settings
        for option, default_value in settings.iteritems():
            if not weechat.config_is_set_plugin(option):
                weechat.config_set_plugin(option, default_value)

        weechat.hook_command(SCRIPT_COMMAND,
                             "URL bar control",
                             "[list | hide | show | toggle | URL]",
                             "   list: list all URL and show URL bar\n"
                             "   hide: hide URL bar\n"
                             "   show: show URL bar\n"
                             "   toggle: toggle showing of URL bar\n",
                             "list %(urlbar_urls)",
                             "urlbar_cmd", "")
        weechat.hook_completion("urlbar_urls", "list of URLs",
                                "urlbar_completion_urls_cb", "")
        weechat.bar_item_new("urlbar_urls", "urlbar_item_cb", "");
        weechat.bar_new("urlbar", "on", "0", "root", "", "top", "horizontal",
                        "vertical", "0", "0", "default", "default", "default", "0",
                        "urlbar_urls");
        weechat.hook_print("", "", "://", 1, "urlbar_print_cb", "")
```

Pour revoir ces <url> taper dans tty X :

```
/urlbar toggle
```

Nota :

Si vous opérez l'installation de l'application weechat-plugins, weechat activé, il vous faudra utiliser dans weechat même la commande de relance pour la mise à jour :

```
 /upgrade
```

Pour l'installation d'un nouveau script seulement, toujours weechat activé, vous pouvez le lancer par

exemple :

```
/python load urlbar.py
```

Configuration manuelle

On peut modifier cela encore ! 😀

Script - Modification

Weechat activé, nous allons d'abord modifier les comportements par defaut de **urlbar.py** en rédigeant dans weechat les commandes suivantes.

La barre garde 10 url au lieu de 5 ainsi :

```
/set plugins.var.python.urlbar.visible_amount 10
```

La barre reste apparente pendant une minute ainsi :

```
/set plugins.var.python.urlbar.visible_seconds 60
```

Nous pouvons aussi désactiver tous ces comportements par défaut pour configurer les nôtres.
Taper :

```
/set plugins.var.python.urlbar.use_popup "off"
```

Ensuite créer un raccourcis clavier, par exemple le raccourcis Alt + L ainsi :

```
/key bind meta-l /urlbar toggle
```

Maintenant, pour faire apparaître et disparaître à volonté les <*url*> des différents chans dans la fenêtre **urlbar**, utiliser le raccourcis créé :

Alt + L

Les liens

- Demarrage Rapide
- Site Weechat officiel
- Scripts disponibles

😀😀😀

1)

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:weechat

Last update: **11/02/2024 11:01**