FreePOPs Manual

Written by Enrico Tassi, Nicola Cocchiaro

Date : 2007/10/2812 : 32 : 22
Revision : 1.41

Document released under the GNU/GPL license.

Contents

1 Introduction **4**
1.1 Usage situations 4
1.2 Features 4
1.3 Plugins 5

2 History **9**

3 FreePOPs configuration file **9**
3.1 Simple load balancing 10

4 FreePOPs command line parameters **11**

5 Email client configuration **13**
5.1 Outlook Express tutorial 14
5.2 Proxy tutorial 17
5.3 Spam/AV tutorial 17
5.3.1 Norton AntiVirus, version 2002 and up 17
5.3.2 Avast! AntiVirus 17
5.3.3 AVG Pro 7 AntiVirus 18
5.3.4 SpamHilator 18
5.3.5 Mailshield Desktop 18
5.3.6 K9 18
5.3.7 SpamTerminator 18
5.3.8 SpamPal 18
5.4 LAN tutorial 18

6 Plugins **19**
6.1 Parameters 19
6.2 abv.lua 19
6.3 aol.lua 20
6.4 davmail.lua 21
6.5 excite.lua 21
6.6 fastmail.lua 22
6.7 gmail.lua 23
6.8 hotmail.lua 24
6.9 juno.lua 26
6.10 kernel.lua 26
6.11 libero.lua 27
6.12 lycos.lua 28
6.13 mail2world.lua 28
6.14 mailcom.lua 29

6.15 monitor.lua 31
6.16 netscape.lua 32
6.17 orange.lua 32
6.18 popforward.lua 33
6.19 softhome.lua 34
6.20 squirrelmail.lua 34
6.21 supereva.lua 35
6.22 tin.lua 35
6.23 tre.lua 36
6.24 updater.lua 37
6.25 yahoo.lua 37
6.26 aggregator.lua 38
6.27 flatnuke.lua 40

7 Creating a plugin **41**
7.1 Plugins overview 42
7.1.1 The interface between the C core and a plugin 42
7.1.2 The interface between a plugin and the C core 43
7.2 The art of writing a plugin (plugins tutorial) 44
7.2.1 (step 1) The skeleton 44
7.2.2 (step 2) The login 45
7.2.3 (step 3) Getting the list of messages 51
7.2.4 (step 4) The common functions 55
7.2.5 (step 5) Deleting messages 56
7.2.6 (step 6) Downloading messages 57
7.2.7 (step 7) Test it 58
7.2.8 (step 8) The so mentioned last part of the tutorial 59

8 Submitting a bug **64**

9 FAQ **65**

10 Authors **67**
10.1 Developers 67

11 Thanks **68**

1 Introduction

FreePOPs is a POP3 daemon plus a LUA interpreter and some extra libraries for HTTP and HTML parsing. Its main purpose is translating local POP3 requests to remote HTTP actions on the supported web-mails, but it is really more flexible: for example there is a plugin to read news from a website as if they were mails in a mailbox. You can easily extend FreePOPs on the fly, without even restarting it; you can add a plugin or modify an existing one simply changing the script file since the plugins are written in LUA and are interpreted on the fly.

1.1 Usage situations

FreePOPs can be useful in some situations, here we give the most obvious ones:

- You are behind a firewall that closes port 110 but you need to read your mail and the web-mail of your mail provider sucks.
- Your mail provider does not allow you to access your mailbox with the POP3 protocol, but only through the web-mail service.
- You prefer looking at your mailbox instead of browsing some websites news.
- You have to develop a POP3 server in less than a week and you want it to be reasonably fast and not so memory consuming.
- You are not a C hacker, but you want to benefit from a fast POP3 server frontend written in C and you want not to waste a month in writing the backend in C. LUA is a really simple and tiny language, one week is enough to learn it in a way that allows you to use it productively.

1.2 Features

FreePOPs is the only software I know with these features:

- POP3 server RFC compliant (not full featured but compliant).
- Portable (written in C and LUA that is written in C, so everything is written in the most portable language around the world).
- Small (in the sense of resources usage) and reasonably fast.
- Extremely extensible on the fly using a simple and powerful language.
- Pretty documented.
- Released under the GNU/GPL license (this means FreePOPs is Free Software).

1.3 Plugins

These are the plugins currently included in FreePOPs:

abv.lua (abv.bg) :
This plugin supports these domains: @abv.bg, @gyuvetch.bg, @gbg.bg

aggregator.lua (RSS/RDF aggregator) :
This plugin supports these domains: @aggregator, @...

aol.lua (aol.com) :
This plugin supports these domains: @aol.com, @aol.com.ar, @aol.fr, @aol.com.mx, @aol.com.au, @aol.de, @aol.com.pr, @aol.com.br, @jp.aol.com, @aol.com.uk, @aol.ca, @aola.com, @aim.com

davmail.lua (DAVMAIL) :
This plugin supports these domains: @lycos.co.uk, @lycos.ch, @lycos.de, @lycos.es, @lycos.it, @lycos.at, @lycos.nl, @spray.se, @jubii.dk

excite.lua (excite) :
This plugin supports these domains: @excite.com, @myway.com

fastmail.lua (fastmail.com) :
This plugin supports these domains: @123mail.org, @150mail.com, @150ml.com, @16mail.com, @2-mail.com, @4email.net, @50mail.com, @airpost.net, @all-mail.net, @bestmail.us, @cluemail.com, @elitemail.org, @emailgroups.net, @emailplus.org, @emailuser.net, @eml.cc, @fastem.com, @fast-email.com, @fastemail.us, @fastemailer.com, @fastest.cc, @fastimap.com, @fastmail.cn, @fastmail.com.au, @fastmail.fm, @fastmail.us, @fastmail.co.uk, @fastmail.to, @fmail.co.uk, @fast-mail.org, @fastmailbox.net, @fastmessaging.com, @fea.st, @f-m.fm, @fmailbox.com, @fmgirl.com, @fmguy.com, @ftml.net, @hail-mail.net, @imap.cc, @imap-mail.com, @imapmail.org, @internet-e-mail.com, @internetemails.net, @internet-mail.org, @internetmailing.net, @jetemail.net, @justemail.net, @letterboxes.org, @mailandftp.com, @mailas.com, @mail-bolt.com, @mailc.net, @mailcan.com, @mail-central.com, @mailforce.net, @mailftp.com, @mailhaven.com, @mailingaddress.org, @mailite.com, @mailmight.com, @mailnew.com, @mail-page.com, @mailsent.net, @mailservice.ms, @mailup.net, @mailworks.org, @ml1.net, @mm.st, @myfastmail.com, @mymacmail.com, @nospammail.net, @ownmail.net, @petml.com, @postinbox.com, @post-pro.net, @proinbox.com, @promessage.com, @realemail.net, @reallyfast.biz, @reallyfast.info, @rushpost.com, @sent.as, @sent.at, @sent.com, @speed-post.net, @speedymail.org, @ssl-mail.com, @swift-mail.com, @the-fastest.net, @theinternetemail.com, @the-quickest.com, @veryfast.biz, @veryspeedy.net, @warpmail.net, @xsmail.com, @yepmail.net, @your-mail.com

flatnuke.lua (flatnuke) :
This plugin supports these domains: @flatnuke, @...

gmail.lua (GMail.com) :
This plugin supports these domains: @gmail.com

hotmail.lua (hotmail.com) :
This plugin supports these domains: @hotmail.com, @msn.com, @webtv.com, @charter.com, @compaq.net, @passport.com, @hotmail.de, @hotmail.it, @hotmail.co.uk, @hotmail.co.jp, @hotmail.fr, @messengeruser.com, @hotmail.com.ar, @hotmail.co.th, @hotmail.com.tr, @milanosemplice.it

juno.lua (juno.com) :
This plugin supports these domains: @netzero.net, @netzero.com, @juno.com

kernel.lua (kernel.org Changelog viewer) :
This plugin supports these domains: @kernel.org, @kernel.org.24, @kernel.org.26

libero.lua (Libero.IT) :
This plugin supports these domains: @libero.it, @inwind.it, @iol.it, @blu.it

lycos.lua (Lycos.IT (this plugin doesn't work!)) :
This plugin supports these domains: @...

mail2world.lua (mail2world.com) :
This plugin supports these domains: @mail2*.com

mailcom.lua (mail.com) :
This plugin supports these domains: @mail.com, @email.com, @iname.com, @cheerful.com, @consultant.com, @europe.com, @mindless.com, @earthling.net, @myself.com, @post.com, @techie.com, @usa.com, @writeme.com, @2die4.com, @artlover.com, @bikerider.com, @catlover.com, @cliffhanger.com, @cutey.com, @doglover.com, @gardener.com, @hot-shot.com, @inorbit.com, @loveable.com, @mad.scientist.com, @playful.com, @poetic.com, @popstar.com, @popstarmail.org, @saintly.com, @seductive.com, @soon.com, @whoever.com, @winning.com, @witty.com, @yours.com, @africamail.com, @arcticmail.com, @asia.com, @australiamail.com, @europe.com, @japan.com, @samerica.com, @usa.com, @berlin.com, @dublin.com, @london.com, @madrid.com, @moscowmail.com, @munich.com, @nycmail.com, @paris.com, @rome.com, @sanfranmail.com, @singapore.com, @tokyo.com, @accountant.com, @adexec.com, @allergist.com, @alumnidirector.com, @archaeologist.com, @chemist.com, @clerk.com, @columnist.com, @comic.com, @consultant.com, @counsellor.com, @deliveryman.com, @diplomats.com, @doctor.com, @dr.com, @engineer.com, @execs.com, @financier.com, @geologist.com, @graphic-designer.com,

@hairdresser.net, @insurer.com, @journalist.com, @lawyer.com, @legislator.com, @lobbyist.com, @minister.com, @musician.org, @optician.com, @pediatrician.com, @presidency.com, @priest.com, @programmer.net, @publicist.com, @realtyagent.com, @registerednurses.com, @repairman.com, @representative.com, @rescueteam.com, @scientist.com, @sociologist.com, @teacher.com, @techie.com, @technologist.com, @umpire.com, @02.to, @111.ac, @123post.com, @168city.com, @2friend.com, @65.to, @852.to, @86.to, @886.to, @aaronkwok.net, @acmilan-mail.com, @allstarstats.com, @amrer.net, @amuro.net, @amuromail.com, @anfieldroad-mail.com, @arigatoo.net, @arsenal-mail.com, @barca-mail.com, @baseball-mail.com, @basketballmail.com, @bayern-munchen.com, @birmingham-mail.com, @blackburnmail.com, @bsdmail.com, @bsdmail.org, @c-palace.com, @celtic-mail.com, @charlton-mail.com, @chelsea-mail.com, @china139.com, @chinabyte.com, @chinahot.net, @chinarichholdings.com, @coolmail.ac, @coventry-mail.com, @cseek.com, @cutemail.ac, @daydiary.com, @dbzmail.com, @derby-mail.com, @dhsmail.org, @dokodemo.ac, @doomo.net, @doramail.com, @e-office.ac, @e-yubin.com, @eracle.com, @eu-mail.net, @everton-mail.com, @eyah.com, @ezagenda.com, @fastermail.com, @femail.ac, @fiorentina-mail.com, @footballmail.com, @forest-mail.com, @freeid.net, @fulham-mail.com, @gaywiredmail.com, @genkimail.com, @gigileung.org, @glay.org, @globalcom.ac, @golfmail.com, @graffiti.net, @gravity.com.au, @hackermail.com, @highburymail.com, @hitechweekly.com, @hkis.org, @hkmag.com, @hkomail.com, @hockey-mail.com, @hollywood-mail.com, @ii-mail.com, @iname.ru, @inboexes.org, @inboxes.com, @inboxes.net, @inboxes.org, @insingapore.com, @intermilan-mail.com, @ipswich-mail.com, @isleuthmail.com, @jane.com.tw, @japan1.org, @japanet.ac, @japanmail.com, @jayde.com, @jcom.ac, @jedimail.com, @joinme.com, @joyo.com, @jpn1.com, @jpol.net, @jpopmail.com, @juve-mail.com, @juventus-mail.com, @juventusmail.net, @kakkoii.net, @kawaiimail.com, @kellychen.com, @keromail.com, @kichimail.com, @kitty.cc, @kittymail.com, @kittymail.net, @lazio-mail.com, @lazypig.net, @leedsmail.com, @leicester-mail.com, @leonlai.net, @linuxmail.org, @liverpoolmail.com, @luvplanet.net, @mailasia.com, @mailjp.net, @mailpanda.com, @mailunion.com, @man-city.com, @manu-mail.com, @marchmail.com, @markguide.
@maxplanet.com, @megacity.com, @middlesbrough-mail.com, @miriamyeung.com, @miriamyeung.com.hk, @myoffice.ac, @nctta.org, @netmarketingcentral.com, @nettalk.ac, @newcastle-mail.com, @nihonjin1.com, @nihonmail.com, @norikomail.com, @norwich-mail.com, @old-trafford.com, @operamail.com, @otakumail.com, @outblaze.net, @outgun.com, @pakistans.com, @pokefan.com, @portugalnet.com, @powerasia.com, @qpr-mail.com, @rangers-mail.com, @realmadrid-mail.com, @regards.com, @ronin1.com, @rotoworld.com, @samilan.com, @searcheuropemail.com, @sexymail.ac, @sheff-wednesday.com, @slonline.net, @smapxsmap.net, @southamptonmail.com, @speedmail.ac, @sports-mail.com, @starmate.com, @sunderland-

mail.com, @sunmail.ac, @supermail.ac, @supermail.com, @surfmail.ac, @surfy.net, @taiwan.com, @talknet.ac, @teddy.cc, @tennis-mail.com, @tottenham-mail.com, @utsukushii.net, @uymail.com, @villa-mail.com, @webcity.ca, @webmail.lu, @welcomm.ac, @wenxuecity.net, @westham-mail.com, @wimbledon-mail.com, @windrivers.net, @wolves-mail.com, @wongfaye.com, @world-mail.ac, @worldweb.ac, @isleuthmail.com, @x-lab.cc, @xy.com.tw, @yan-keeman.com, @yyhmail.com, @verizonmail.com, @lycos.com, @cyberdude.com, @mail.org, @italymail.com, @mexico.com, @india.com, @u2club.com, @royal.net

monitor.lua (monitor) :
This plugin supports these domains: @monitor

netscape.lua (netscape.net) :
This plugin supports these domains: @netscape.net

orange.lua (Orange (ex Wanadoo)) :
This plugin supports these domains: @fsmail.net, @wanadoo.nl, @orange.nl, @bedrijfsnaam.nl

popforward.lua (POPforward) :
This plugin supports these domains: @...

softhome.lua (softhome.net) :
This plugin supports these domains: @softhome.net

squirrelmail.lua (SquirrelMail) :
This plugin supports these domains: @...

supereva.lua (Supereva) :
This plugin supports these domains: @supereva.it, @supereva.com, @freemail.it, @freeweb.org, @mybox.it, @superdada.com, @cicciociccio.com, @mp4.it, @dadacasa.com, @clarence.com, @concento.it, @dada.net

tin.lua (Tin.IT) :
This plugin supports these domains: @tin.it, @virgilio.it, @alice.it, @tim.it, @atlantide.it

tre.lua (Tre) :
This plugin supports these domains: @tre.it, @three.com.au

updater.lua (updater) :
This plugin supports these domains: @updater

yahoo.lua (yahoo.com) :
This plugin supports these domains: @yahoo.com, @yahoo.ie, @yahoo.it, @yahoo.ca, @rocketmail.com, @yahoo.com.ar, @yahoo.co.in, @yahoo.co.id,

@yahoo.com.tw, @yahoo.co.uk, @yahoo.com.cn, @yahoo.es, @yahoo.de, @talk21.com, @btinternet.com, @yahoo.com.au, @yahoo.co.nz, @ymail.com, @yahoo.in

2 History

FreePOPs was not born from scratch. A similar project (only in the main usage situation) is LiberoPOPs.

The ancestor of FreePOPs is completely written in C for some uninteresting reasons. LiberoPOPs supports "plugins" but in a more static and complex way. The POP3 server frontend could be attached to a backend written in C, this means you have to recompile and restart LiberoPOPs each time to change a line in a plugin. Another interesting point is that LiberoPOPs was created from scratch in a really short time (you have to be Italian and use a `@libero.it` mail address to understand why), this means it was born with a lot of bugs and FIX-ME in the code.

The LiberoPOPs project had a quick success, because everybody needed it, and this means we had a lot of users. In the opensource (and also Linux) philosophy you have to release frequently and this was exactly what we did: we used to release every two days. We were working not with Unix users, nor hackers, but mostly with Win32 users. Suddenly we realized that they were lazy/bored of updating the software every 2 days. The ugly Win-world has taught them that software auto-updates, auto-install and even auto-codes probably.

We tried to solve this pulling out of the C engine most of the change-prone code, but this was really hard since C is not thought to do this. After LiberoPOPs had stabilized we started to think how to solve this.

A scripting/interpreted embedded language seemed to me a nice choice and after a long search on the net and on the newsgroup of my university I found LUA.. This is not the place for telling the world how good this small language is and I won't talk more about it here. Integrating the LUA interpreter in LiberoPOPs was not so hard and FreePOPs is the result. Now it is really easier to write/test a plugin and (even if it is not implemented yet) an auto-update facility is really easy to code since there is no need to recompile the C core in most cases.

3 FreePOPs configuration file

FreePOPs doesn't really need a configuration. Most users shouldn't change the configuration file. In case you are a developer or a really curious user the

configuration file is `config.lua`, placed in the program directory under win32 or in `/etc/freepops/` in a posix environment.

Later you will learn that plugins are associated with a mail address domain, and some of these plugins are aliased to other domains to make it easier to fetch some news from some sites. Read the plugin documentation for more info about them, and maybe send as a mail with your new alias if you want it to be integrated in the next FreePOPs release.

Since version 0.0.11 the `config.lua` file has a policy section. In this section you can ban or allow classes of mail addresses. This may be useful to network administrators.

3.1 Simple load balancing

A simple way to run multiple instances of FreePOPs to serve the same pool of users is to use an additional instance of FreePOPs as a proxy that chooses to which FreePOPs instance each connection has to be forwarded. We will call the the proxy instance of FreePOPs master and the other running instances slaves. Consider also the simple case in which the number of concurrent connections accepted by each slave (parameter `-t`) is five and the number of slaves is three. you may want to run the master instance with the following command line:

```
freepopsd -c balance.lua -t 15 -p 110 -b 0.0.0.0
```

The configuration file `balance.lua` is

```
freepops.MODULES_MAP[".*"] = {
    name = "popforward.lua",
    regex = true,
    args = {
        host = function(mailaddress)
            local name = freepops.get_name(mailaddress)
            local slaves = {
                { rex = '^[0-9]',    host = "localhost", port = 2000 },
                { rex = '^[a-lA-l]', host = "localhost", port = 2001 },
                { rex = '^[m-zM-Z]', host = "localhost", port = 2002 },
            }
            local host, port = slaves[1].host, slaves[1].port -- defaults
            for _, slave in ipairs(slaves) do
                if string.match(name, slave.rex) then
                    host = slave.host
                    port = slave.port
                    break
                end
```

```
            end
            return host, port
        end
    }
}
freepops.MODULES_PREFIX = {
    os.getenv("FREEPOPSLUA_PATH_UPDATES") or "./",
    os.getenv("FREEPOPSLUA_PATH") or "./",
    "./lua/", "./", "./src/lua/", "./modules/include/", "./modules/lib/"}
freepops.MODULES_CPREFIX = {
    os.getenv("FREEPOPSLUA_CPATH") or "./",
    "./c/", "./", "./updater-ui/fltk/" }
freepops.MODULES_PREFIX_UNOFFICIAL = {
    os.getenv("FREEPOPSLUA_PATH_UNOFFICIAL") or "./",
    "./src/lua_unofficial/", }
```

The whole file is reported for completeness, but the last part is the standard one. The host parameter to the popforward plugin is a function returning different host and port according to the email address taken in input. This simple balancing mechanism is supported from version 0.2.6. Note that the master instance has to keep a connection open for every client.

4 FreePOPs command line parameters

The real FreePOPs configuration is made trough command line arguments. They are described in depth in the man page in Unix environments and below. Keep in mind that in normal usage situations it's not necessary to use any of these, but if you have special needs it's useful to use the following list as reference:

-p <port>, --port <port> By default FreePOPs binds on port 2000. To alter this behavior just use this switch.

-t <num>, --threads <num> FreePOPs is able to manage multiple connections, up to *num*. Default is 5.

-b addr, --bind addr Binds over *addr* instead INADDR_ANY (0.0.0.0). *addr* must be a character string containing an IPv4 network address in the dotted-quad format, "ddd.ddd.ddd.ddd" or a host name.

-l logfacility, --logmode logfacility Can be used to specify the logging facility. *logfacility* can be "stdout" for stdout (the default), "syslog" to use the logging daemon or a valid filename to log to this file.

-d, - -daemonize Moves the process to background releasing the tty.

-n, - -no-pid-file Do not generate the file containing the process' pid in /ver/run/.

-P <host>:<port>, - -proxy <host>:<port> To tell FreePOPs which is your HTTP proxy. If *port* is not set then the default 8080 is used.

-A <username>:<password>, - -auth <username>:<password> For proxies with basic authentication, to specify username and password. Must be used with *-P* or its long form.

-u name, - -useragent name Use this useragent in http connections. The default is "Firefox/0.8". A valid example is mozilla's "Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.5) Gecko/20031024 Debian/1.5-2".

-s user.group, - -suid user.group This option is used to make freepopsd drop root privileges after binding. If you run it as a normal user there is no need to use this option. *(Not used under Windows)*

-k, - -kill Terminates a running FreePOPs program. *(Not used under Windows)*

-x pluginfile, - -toxml file Prints on standard output the XML description of the plugin or module.

-e scriptfile args..., - -execute scriptfile args... This is a full bloated LUA interpreter, the executed script has access to all freepops libraries. The interpreter calls the main function that must get a table of strings and return an integer. The arguments passed to freepopsd after the script file name are put inside the table argument. The return value is returned from the interpreter.

-c, - -conffile file Users the specified configuration file instead of looking in default paths like /etc/freepops/config.lua, ./config.lua and /usr/share/frepops/lua/c

- -statistics-all Enable all statistics. Results can be viewed with the monitor plugin, either with an account like foo@monitor?command=stats or with freepopsd -e monitor host port password command.

- -statistics-session-created Enables statistics regarding threads created to run a plugin. See the documentation of - -statistics-all for an explanation of how to read that statistics.

- -statistics-session-ok Enables statistics regarding sessions ended successfully. See the documentation of - -statistics-all for an explanation of how to read that statistics.

- -statistics-session-err Enables statistics regarding sessions ended with an error. See the documentation of - -statistics-all for an explanation of how to read that statistics.

- -statistics-connection-established Enables statistics regarding connections accepted. See the documentation of - -statistics-all for an explanation of how to read that statistics.

- -statistics-cookies Enables statistics regarding persistently stored data (usually cookies). See the documentation of - -statistics-all for an explanation of how to read that statistics.

- -statistics-pwd-file file Data collected by the statistics mechanism can be read using the monitor plugin. If a password file is not specified, no password is set and everybody connection to the freepops daemon can read such data. Write your password in a text file with no additional end-of-line to restrict access to that data. The password file is read before dropping privileges (on unix).

- -fpat authtype, - -force-proxy-auth-type authtype To forse a specific proxy auth method. Accepted values are: ntlm, basic, digest e gss.

- -no-icon To disable the win32 systray icon (windows only).

-h, - -help Prints the usage message.

-v, - -verbose, -w, - -veryverbose This tells FreePOPs to log some interesting info for bug reporting.

In posix environments like Debian GNU/Linux you can start FreePOPs at boot time as a standard service. In this case the command line switches are stored in `/etc/default/freepops`, in some rpm based systems you should find the same file as `/etc/sysconfig/freepops`.

5 Email client configuration

To configure your email client you must change the pop3 server settings. Usually you must use localhost as the pop3 host name, and 2000 as the pop3 port. In case you install FreePOPs in another computer of your LAN, you should use the host's name instead of localhost, while in case you changed the default port with the `-p` switch you will have to use that same port in your email client. You always have to use a full email address as username, for example `something@libero.it` instead of only `something`. This is because FreePOPs chooses the plugin to load by looking at your username. Later we will present all the plugins and their associated domains, and how to create an on-the-fly binding between a mail address and a domain.

5.1 Outlook Express tutorial

Here's a tutorial for configuring Outlook Express in a Windows environment. Other mail clients should be configured similarly.

- From the tools menu choose the **Account...** item (see figure 1)



Figure 1: main

- Select your account and click on **Properties** (see figure 2)

- In the Server tab type in **Incoming mail** the name of the computer where you started FreePOPs, usually *localhost*. The **Account name** must be your complete email address, followed by the domain name your email belongs to, for example `username@domain.com` (see figure 3).

- In the **Advanced settings** tab type in **Incoming mail** the port number, that is *2000* if you accepted our settings. ***De-select*** *This server requires a secure connection (SSL)* (see figure 4).

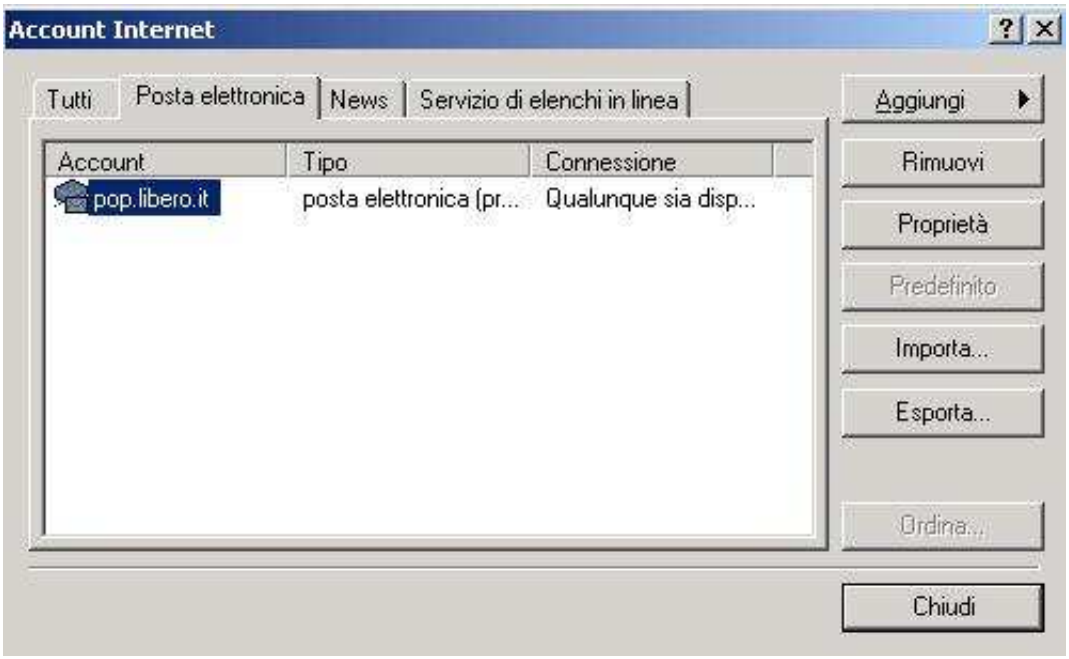


Figure 2: settings

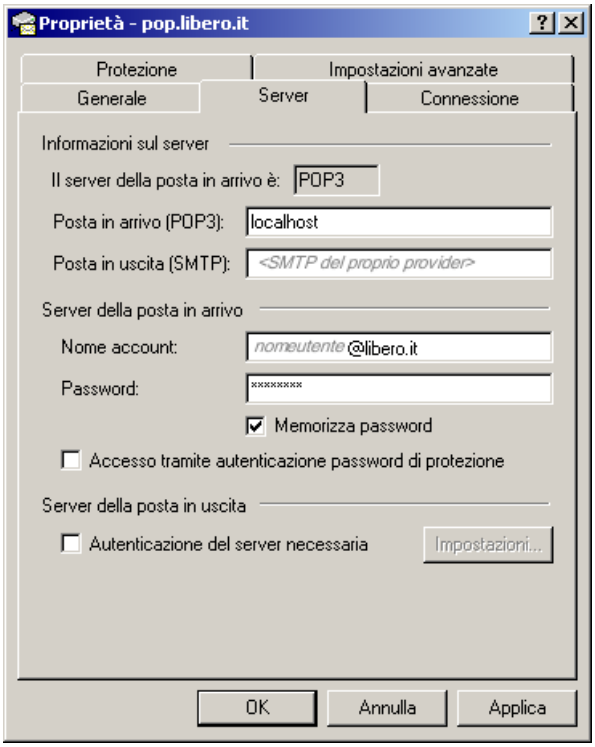


Figure 3: server

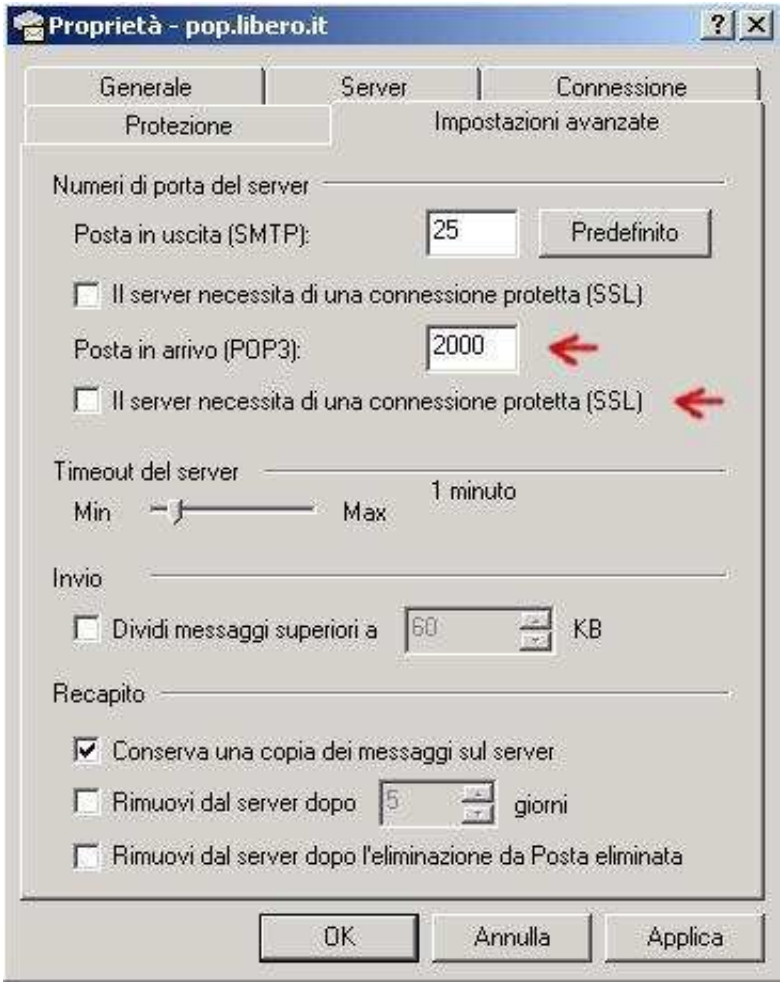


Figure 4: advanced

5.2 Proxy tutorial

FreePOPs is able to use HTTP proxy servers. If you don't know what they are or if there's no proxy in your local network then you may skip this page, as the operations described herein will be useless to you.

In order to use a HTTP proxy, FreePOPs supports the `-P` option, or the equivalent long version `--proxy`, to specify the address and port of the proxy separated by : (colon), for example `-P proxy.localnet.org:8080` or `-P 192.168.1.1` are valid choices. If no port number is specified then `8080` will be used as a default value.

If authentication is necessary in order to use a proxy, use also the `-A username:password` option.

Remember that the values specified with the `-P` option have precedence over any other value obtained by the operating system in use.

In POSIX environments it's possible to use a proxy also using some environment variables.

The environment variables that will be used are, in order of precedence, `HTTP_PROXY`, `http_proxy`, `PROXY` and `proxy`.

The current implementation supports some proxy authentication methods, and some of them require the SSL version of FreePOPs.

5.3 Spam/AV tutorial

Several Windows users, in collaboration with the LiberoPOPs team have created a tutorial for antispam and antivurus software. This tutorial is for FreePOPs too.

5.3.1 Norton AntiVirus, version 2002 and up

It is necessary to have FreePOPs listen on port 110 by means of the `-p` option and then set your email client so that it receives mail on port 110. To change FreePOPs options, read the FAQ "How do I change FreePOPs's command line switches?" question.

5.3.2 Avast! AntiVirus

In your email client, change the username like this: `email@address#localhost:2000` Inside the email client options, set the POP3 server port number to 110, instead of what explained in the previous tutorials.

5.3.3 AVG Pro 7 AntiVirus

In your email client, modify the POP3 port number to 5300, leave unmodified the username and server (`email@address` and `localhost`). In AVG, enter "Properties > Servers > Create a POP3 mail server (server type)", in connection set Fixed host: 127.0.0.1:2000 and Local port: 5300

5.3.4 SpamHilator

Configure your email client with the following parameters: POP3 server (incoming mail): localhost POP3 server port: 110 Username: `localhost&email@address&2000`

5.3.5 Mailshield Desktop

In Mailshield Desktop, choose "Edit mail account", choosing the account you want to modify. In "Account name" and "Email address" type your complete email address. Then choose "Access", in "type of Email server" use "POP3 mail account", while in "incoming mail server" type 127.0.0.1. It may be useful to select the option "Use relaxed timeouts with this email server".

5.3.6 K9

Set up your email client to use 9999 as POP3 server port. Leave localhost as server name. Then use `localhost/2000/email@address` as username.

5.3.7 SpamTerminator

Configure your email client with the following parameters: POP3 server (incoming mail): localhost POP3 server port: 8110 Username: `email@address#localhost` Then start FreePOPs with the option `-p` 110.

5.3.8 SpamPal

Configure your email client with the following parameters: POP3 server (incoming mail): localhost POP3 server port: 110 Username: `email@address@localhost:2000`

5.4 LAN tutorial

How to use FreePOPs as a server in a computer network (Windows-oriented tutorial).

A LAN is composed of 2 computers (or more, but from 2 to 100 it is the same). We'll call *Sola* the server and *Cucco* the client. FreePOPs will start on *Sola* with these options:

`freepopsd.exe -b 0.0.0.0 -p 110`
that means that FreePOPs will bind on 0.0.0.0 (all network interfaces, offering the service to all) at port 110, the default POP3 port. Now we configure the mail client on Cucco, setting the POP3 server to *Sola* and the server port to *110*. Now, consider *Sola* has a monitor too and we want to read some mail from here. We have to set the server to *localhost* and the port to *110*.

By default under Windows FreePOPs binds on `127.0.0.1` offering the service only to the local computer, so the `-b` switch is really important here.

6 Plugins

Here we give a detailed description of each plugin, but before starting we explain the general way of passing special arguments to plugins (see the specific plugin description for a detailed description of the accepted parameters).

6.1 Parameters

Each plugin can receive parameters passed as an addon to username. The following username is for the `popforward.lua` plugin:
`gareuselesinge@mydomain.xx?host=pop.mydomain.xx&port=110`
Since you may use some antispam proxy or other program that may handle your username and may dislike the `?` character you may use a space instead of it. Every following character that is not a letter or a number needs to be escaped. So they have to be written as `%xx`, where `xx` is the hexadecimal code of the corresponding character (exacly as it happens in URL). The space character could also be substituted simply by a plus characted `+`. For example, if you have to assign the value `"My Messages"` to the `folder` parameter, you need to write `folder=My+Messages`.Take a look to the Appendix, to find more informations about hexadecimal codes.

Another way of hacking with the username is the on-the-fly domain-plugin binding. You may find useful to say: “I want to use plugin X for domain Y without changing the config.lua file”. In this case you have to use the plugin name (for example `popforward.lua`) as the domain name an probably you will have to pass some arguments to the plugin using the procedure previously described. This is an example:
`gareuselesinge@popforward.lua?host=pop.mymailsite.xx&port=110`
Remember that in the case of the use of on-the-fly bindings there will be no default arguments, thus `port=110` can’t be omitted as in the previous example.

6.2 abv.lua

Name: abv.bg

Version: 0.1.6

Requires: FreePOPs 0.0.97

License: GNU/GPL

Available at: http://freepops.sourceforge.net/download.php?contrib=abv.lua

Homepage: http://freepops.sourceforge.net/

Author: Russell Schwager <russell822 (at) yahoo (.) com>

Domains: @abv.bg, @gyuvetch.bg, @gbg.bg

Domain(regex):

Description: To use this plugin you have to use your full email address as the username and your real password as the password. For support, please post a question to the forum instead of emailing the author(s).

6.3 aol.lua

Name: aol.com

Version: 0.1.3

Requires: FreePOPs 0.2.0

License: GNU/GPL

Available at: http://freepops.sourceforge.net/download.php?module=aol.lua

Homepage: http://freepops.sourceforge.net/

Author: Russell Schwager <russells (at) despammed (.) com>

Domains: @aol.com, @aol.com.ar, @aol.fr, @aol.com.mx, @aol.com.au, @aol.de, @aol.com.pr, @aol.com.br, @jp.aol.com, @aol.com.uk, @aol.ca, @aola.com, @aim.com

Domain(regex):

Description: To use this plugin you have to use your full email address as the username and your real password as the password.

6.4 davmail.lua

Name: DAVMAIL

Version: 0.0.6

Requires: FreePOPs 0.2.0

License: GNU/GPL

Available at: http://www.freepops.org/download.php?module=davmail.lua

Homepage: http://www.freepops.org/

Author: Enrico Tassi <gareuselesinge (at) users (.) sourceforge (.) net>

Domains: @lycos.co.uk, @lycos.ch, @lycos.de, @lycos.es, @lycos.it, @lycos.at, @lycos.nl, @spray.se, @jubii.dk

Domain(regex):

Description: This plugin implements the HTTPMAIL protocol.

Limitation for Jubii.dk: it is not possible to "leave a copy of the messages on server" since after they've been seen once they disappear (but are not deleted, since via web you can manage them). So, if you don't want to use the webmail to empty your mailbox, don't mark the "leave a copy..." option in your account configuration.

Parameter:

folder The folder you want to read. Default is inbox.

6.5 excite.lua

Name: excite

Version: 0.0.5a

Requires: FreePOPs 0.0.99

License: GNU/GPL

Available at: http://www.freepops.org/download.php?contrib=excite.lua

Homepage: http://www.freepops.org/en/viewplugins.php

Author: TheMarco <themarco (at) fsmail (.) net>

Domains: @excite.com, @myway.com

Domain(regex):

Description: Excite webmail plugin. Use your full email address as the username and your real password as the password. For support, please post your questions to the forum.

Parameter:

folder The folder to interact with. Default is Inbox, other values are: Bulk, Sent, Trash, Drafts or user defined folder.

6.6 fastmail.lua

Name: fastmail.com

Version: 0.0.3c

Requires: FreePOPs 0.2.0

License: GNU/GPL

Available at: http://www.freepops.org/download.php?module=fastmail.lua

Homepage: http://www.freepops.org/

Author: Russell Schwager <russell822 (at) yahoo (.) com>

Domains: @123mail.org, @150mail.com, @150ml.com, @16mail.com, @2-mail.com, @4email.net, @50mail.com, @airpost.net, @allmail.net, @bestmail.us, @clubemail.com, @elitemail.org, @emailgroups.net, @emailplus.org, @emailuser.net, @eml.cc, @fastem.com, @fast-email.com, @fastemail.us, @fastemailer.com, @fastest.cc, @fastimap.com, @fastmail.cn, @fastmail.com.au, @fastmail.fm, @fastmail.us, @fastmail.co.uk, @fastmail.to, @fmail.co.uk, @fast-mail.org, @fastmailbox.net, @fastmessaging.com, @fea.st, @f-m.fm, @fmailbox.com, @fmgirl.com, @fmguy.com, @ftml.net, @hailmail.net, @imap.cc, @imapmail.com, @imapmail.org, @internet-e-mail.com, @internetemails.net, @internetmail.org, @internetmailing.net, @jetemail.net, @justemail.net, @letterboxes.org, @mailandftp.com, @mailas.com, @mailbolt.com, @mailc.net, @mailcan.com, @mail-central.com, @mailforce.net, @mailftp.com, @mailhaven.com, @mailingaddress.org, @mailite.com, @mailmight.com, @mailnew.com, @mail-page.com, @mailsent.net, @mailservice.ms, @mailup.net, @mailworks.org, @ml1.net, @mm.st, @myfastmail.com, @mymacmail.com, @nospammail.net, @ownmail.net, @petml.com, @postinbox.com, @postpro.net, @proinbox.com, @promessage.com, @realemail.net, @reallyfast.biz, @reallyfast.info, @rushpost.com,

@sent.as, @sent.at, @sent.com, @speedpost.net, @speedymail.org, @ssl-mail.com, @swift-mail.com, @the-fastest.net, @theinternetemail.com, @the-quickest.com, @veryfast.biz, @veryspeedy.net, @warpmail.net, @xsmail.com, @yepmail.net, @your-mail.com

Domain(regex):

Description: This plugin lets you download mail from fastmail. To use this plugin you have to use your full email address as the username and your real password as the password. For support, please post a question to the forum instead of emailing the author(s).

Parameters:

folder The folder you want to interact with. Default is Inbox, other values are: Junk, Trash, Draft, Sent.

emptytrash Parameter is used to force the plugin to empty the trash when it is done pulling messages. Set the value to 1.

6.7 gmail.lua

Name: GMail.com

Version: 0.0.54

Requires: FreePOPs 0.2.0

License: GNU/GPL

Available at: http://www.freepops.org/download.php?module=gmail.lua

Homepage: http://www.freepops.org/

Authors: Rami Kattan <rkattan (at) gmail (.) com>, EoinK <eoin.pops (at) gmail (.) com>

Domain: @gmail.com

Domain(regex):

Description: This is the webmail support for @gmail.com mailboxes.

To use this plugin you have to use your full email address as the user name and your real password as the password.

Adding some parameters at the end of the username gives the ability to download email from different folder and/or labels, and export the contacts in CSV format. Check "Supported parameters" for more details about the available parameters.

Note:

When the email client issues the command to delete some messages (because in its options it is set to delete messages from the server [after x days]), if checked the inbox folder, the email will be moved to the archive (folder "all"), while if you checked the spam folder, the email will be moved to the trash folder, else it will only be marked as read.

Parameters:

folder Used for selecting the folder to operate on (inbox is the default one).

The standard folders are: inbox, starred, sent, all, spam, trash.

Here is an example of a username to get the email from the starred folder:

foo@gmail.com?folder=starred

If you created custom labels in gmail, you can access them using the label parameter label=name.

maxmsgs Parameter is used to force the plugin to only download a maximum number of messages.

enableimap Parameter is used to turn on IMAP for the account. If set to 1, the plugin will enable IMAP access on the user's account during the QUIT command.

enableforward Parameter is turn on forwarding for the account. If set with an email address, the settings will be updated to enable the user's settings to forward messages to the supplied address.

label Used for selecting the labels to operate on.

Here is an example of a username to get the email from the label Friends:

foo@gmail.com?label=Friends

act Possible values:

- export: Exports your gmail contacts into a file called gmail_contacts_export.csv that will be saved in your home (Unix) or in the My Documents directory (Windows), that can be imported into your email client.

6.8 hotmail.lua

Name: hotmail.com

Version: 0.2.20081219

Requires: FreePOPs 0.2.8

License: GNU/GPL

Available at: http://www.freepops.org/download.php?module=hotmail.lua

Homepage: http://www.freepops.org/

Authors: Russell Schwager <russell822 (at) yahoo (.) com>, D. Milne <drmilne (at) safe-mail (.) net>, Peter Collingbourne <pcc03 (at) doc (.) ic (.) ac (.) uk»

Domains: @hotmail.com, @msn.com, @webtv.com, @charter.com, @compaq.net, @passport.com, @hotmail.de, @hotmail.it, @hotmail.co.uk, @hotmail.co.jp, @hotmail.fr, @messengeruser.com, @hotmail.com.ar, @hotmail.co.th, @hotmail.com.tr, @milanosemplice.it

Domain(regex):

Description: This plugin lets you download mail from @hotmail.com and similar mailboxes. To use this plugin you have to use your full email address as the username and your real password as the password. For support, please post a question to the forum instead of emailing the author(s).

Parameters:

folder The folder you want to interact with. Default is Inbox.

emptyjunk Parameter is used to force the plugin to empty the junk folder when it is done pulling messages. Set the value to 1.

emptytrash Parameter is used to force the plugin to empty the trash when it is done pulling messages. Set the value to 1.

markunread Parameter is used to have the plugin mark all messages that it pulls as unread. If the value is 1, the behavior is turned on.

maxmsgs Parameter is used to force the plugin to only download a maximum number of messages.

domain Parameter is used to override the domain in the email address. This is used so that users don't need to add a mapping to config.lua for a hosted hotmail account.

keepmsgstatus Parameter is used to maintain the status of the message in the state it was before being pulling. If the value is 1, the behavior is turned on and will override the markunread flag..

6.9 juno.lua

Name: juno.com

Version: 0.1.20081222

Requires: FreePOPs 0.2.0

License: GNU/GPL

Available at: http://www.freepops.org/download.php?module=juno.lua

Homepage: http://www.freepops.org/

Author: Russell Schwager <russell822@yahoo.com>

Domains: @netzero.net, @netzero.com, @juno.com

Domain(regex):

Description: This is the webmail support for @juno.com, @netzero.net and @netzero.com mailboxes. To use this plugin you have to use your full email address as the user name and your real password as the password.

Parameters:

folder Parameter is used to select the folder (Inbox is the default) that you wish to access. The folders that are available are the standard Yahoo folders, called Inbox, Draft, Sent, Junk Mail and Trash. For user defined folders, use their name as the value.

emptytrash Parameter is used to force the plugin to empty the trash when it is done pulling messages.

resetheaders Parameter is used to force the plugin to turn off full headers when it is done pulling messages.

noattachments Parameter is used to force the plugin to skip attachments. To turn this on, set the value to 1.

6.10 kernel.lua

Name: kernel.org Changelog viewer

Version: 0.0.4

Requires: FreePOPs 0.2.0

License: GNU/GPL

Available at: http://www.freepops.org/download.php?module=kernel.lua

Homepage: http://www.freepops.org/

Author: Simone Vellei <simone_vellei (at) users (.) sourceforge (.) net>

Domains: @kernel.org, @kernel.org.24, @kernel.org.26

Domain(regex):

Description: This plugin helps in staying up to date with the Linux kernel releases. http://kernel.org is the official page with the Linux kernel releases, each with its ChangeLog. You should use something@kernel.org to receive news about every tree, something@kernel.24 or something@kernel.org.26 for a specific tree. Password is not used, type a random string.

6.11 libero.lua

Name: Libero.IT

Version: 0.2.18

Requires: FreePOPs 0.2.0

License: GNU/GPL

Available at: http://www.freepops.org/download.php?module=libero.lua

Homepage: http://www.freepops.org/

Author: Enrico Tassi <gareuselesinge (at) users (.) sourceforge (.) net>

Domains: @libero.it, @inwind.it, @iol.it, @blu.it

Domain(regex):

Description: This plugin is for italian users only.

Parameter:

folder

6.12 lycos.lua

Name: Lycos.IT (this plugin doesn't work!)

Version: 0.0.5

Requires: FreePOPs 0.2.0

License: GNU/GPL

Available at: ——-

Homepage: ———

Author: — <—->

Domain: @...

Domain(regex):

Description: this plugin cant' do a delete and retr/top. this is half done since now the lycos webmail is served by the owa.lua plugin.

Parameter:

–name– –desc–

6.13 mail2world.lua

Name: mail2world.com

Version: 0.0.2g

Requires: FreePOPs 0.2.0

License: GNU/GPL

Available at: http://freepops.sourceforge.net/download.php?module=mail2world.lua

Homepage: http://freepops.sourceforge.net/

Author: Russell Schwager <russells (at) despammed (.) com>

Domain:

Domain(regex): @mail2*.com

Description: To use this plugin you have to use your full email address as the username and your real password as the password. For support, please post a question to the forum instead of emailing the author(s).

Parameters:

folder The folder you want to interact with. Default is Inbox.

nossl Parameter is used to force the plugin to not use ssl in its session. Set the value to 1 to enable this feature.

6.14 mailcom.lua

Name: mail.com

Version: 0.1.20081128

Requires: FreePOPs 0.2.0

License: GNU/GPL

Available at: http://www.freepops.org/download.php?module=mailcom.lua

Homepage: http://www.freepops.org/

Author: Russell Schwager <russell822 (at) yahoo (.) com>

Domains: @mail.com, @email.com, @iname.com, @cheerful.com, @consultant.com, @europe.com, @mindless.com, @earthling.net, @myself.com, @post.com, @techie.com, @usa.com, @writeme.com, @2die4.com, @artlover.com, @bikerider.com, @catlover.com, @cliffhanger.com, @cutey.com, @doglover.com, @gardener.com, @hot-shot.com, @inorbit.com, @loveable.com, @mad.scientist.com, @playful.com, @poetic.com, @popstar.com, @popstarmail.org, @saintly.com, @seductive.com, @soon.com, @whoever.com, @winning.com, @witty.com, @yours.com, @africamail.com, @arcticmail.com, @asia.com, @australiamail.com, @europe.com, @japan.com, @samerica.com, @usa.com, @berlin.com, @dublin.com, @london.com, @madrid.com, @moscowmail.com, @munich.com, @nycmail.com, @paris.com, @rome.com, @sanfranmail.com, @singapore.com, @tokyo.com, @accountant.com, @adexec.com, @allergist.com, @alumnidirector.com, @archaeologist.com, @chemist.com, @clerk.com, @columnist.com, @comic.com, @consultant.com, @counsellor.com, @deliveryman.com, @diplomats.com, @doctor.com, @dr.com, @engineer.com, @execs.com, @financier.com, @geologist.com, @graphic-designer.com, @hairdresser.net, @insurer.com, @journalist.com, @lawyer.com, @legislator.com, @lobbyist.com, @minister.com, @musician.org, @optician.com, @pediatrician.com, @presidency.com, @priest.com, @programmer.net, @publicist.com, @realtyagent.com, @registerednurses.com, @repairman.com, @representative.com, @rescueteam.com, @scientist.com, @sociologist.com, @teacher.com, @techie.com, @technologist.com, @umpire.com, @02.to, @111.ac, @123post.com, @168city.com,

@2friend.com, @65.to, @852.to, @86.to, @886.to, @aaronkwok.net, @acmilan-mail.com, @allstarstats.com, @amrer.net, @amuro.net, @amuromail.com, @anfieldroad-mail.com, @arigatoo.net, @arsenal-mail.com, @barca-mail.com, @baseball-mail.com, @basketball-mail.com, @bayern-munchen.com, @birmingham-mail.com, @blackburn-mail.com, @bsdmail.com, @bsdmail.org, @c-palace.com, @celtic-mail.com, @charlton-mail.com, @chelsea-mail.com, @china139.com, @chinabyte.com, @chinahot.net, @chinarichholdings.com, @coolmail.ac, @coventry-mail.com, @cseek.com, @cutemail.ac, @daydiary.com, @dbz-mail.com, @derby-mail.com, @dhsmail.org, @dokodemo.ac, @doomo.net, @doramail.com, @e-office.ac, @e-yubin.com, @eracle.com, @eu-mail.net, @everton-mail.com, @eyah.com, @ezagenda.com, @fastermail.com, @fe-mail.ac, @fiorentina-mail.com, @football-mail.com, @forest-mail.com, @freeid.net, @fulham-mail.com, @gaywiredmail.com, @genkimail.com, @gigileung.org, @glay.org, @globalcom.ac, @golf-mail.com, @graffiti.net, @gravity.com.au, @hackermail.com, @highbury-mail.com, @hitechweekly.com, @hkis.org, @hkmag.com, @hkomail.com, @hockey-mail.com, @hollywood-mail.com, @ii-mail.com, @iname.ru, @inboexes.org, @inboxes.com, @inboxes.net, @in-boxes.org, @insingapore.com, @intermilan-mail.com, @ipswich-mail.com, @isleuthmail.com, @jane.com.tw, @japan1.org, @japanet.ac, @japanmail.com, @jayde.com, @jcom.ac, @jedimail.com, @joinme.com, @joyo.com, @jpn1.com, @jpol.net, @jpopmail.com, @juve-mail.com, @juventus-mail.com, @juven-tusmail.net, @kakkoii.net, @kawaiimail.com, @kellychen.com, @keromail.com, @kichimail.com, @kitty.cc, @kittymail.com, @kittymail.net, @lazio-mail.com, @lazypig.net, @leeds-mail.com, @leicester-mail.com, @leonlai.net, @lin-uxmail.org, @liverpool-mail.com, @luvplanet.net, @mailasia.com, @mailjp.net, @mailpanda.com, @mailunion.com, @man-city.com, @manu-mail.com, @march-mail.com, @markguide.com, @maxplanet.com, @megacity.com, @middlesbrough-mail.com, @miriamyeung.com, @miriamyeung.com.hk, @myoffice.ac, @nctta.org, @netmarketingcentral.com, @nettalk.ac, @newcastle-mail.com, @nihon-jin1.com, @nihonmail.com, @norikomail.com, @norwich-mail.com, @old-trafford.com, @operamail.com, @otakumail.com, @outblaze.net, @outgun.com, @pakistans.com, @pokefan.com, @portugalnet.com, @powerasia.com, @qpr-mail.com, @rangers-mail.com, @realmadrid-mail.com, @regards.com, @ronin1.com, @rotoworld.com, @samilan.com, @searcheuropemail.com, @sexymail.ac, @sheff-wednesday.com, @slonline.net, @smapxsmap.net, @southampton-mail.com, @speedmail.ac, @sports-mail.com, @starmate.com, @sunderland-mail.com, @sunmail.ac, @supermail.ac, @supermail.com, @surfmail.ac, @surfy.net, @taiwan.com, @talknet.ac, @teddy.cc, @tennis-mail.com, @tottenham-mail.com, @utsukushii.net, @uymail.com, @villa-mail.com, @webcity.ca, @webmail.lu, @welcomm.ac, @wenxuecity.net, @westham-mail.com, @wimbledon-mail.com, @windrivers.net, @wolves-mail.com, @wongfaye.com, @world-mail.ac, @worldweb.ac, @isleuthmail.com, @x-lab.cc, @xy.com.tw, @yan-keeman.com, @yyhmail.com, @verizonmail.com, @lycos.com, @cyberdude.com,

@mail.org, @italymail.com, @mexico.com, @india.com, @u2club.com, @royal.net

Domain(regex):

Description: This is the webmail support for @mail.com and all its other domain mailboxes. To use this plugin you have to use your full email address as the user name and your real password as the password.

Parameters:

folder Parameter is used to select the folder (Inbox is the default) that you wish to access. The folders that are available are the standard folders, called INBOX, Drafts, SENT, and Trash. For user defined folders, use their name as the value.

emptytrash Parameter is used to force the plugin to empty the trash when it is done pulling messages.

setoptionoverride Parameter is used to tell the plugin not to change the mail options on the mail.com website. If you use this option, you must have full headers enabled in your options. If the value is 1, the behavior is turned on.

usemailcomloginpage Parameter is used to tell the plugin to use the login page on mail.com instead of trying to figure it out by the domain. If the value is 1, the behavior is turned on.

loginpage Parameter is used to tell the plugin which login page to use.

6.15 monitor.lua

Name: monitor

Version: 0.0.1

Requires: FreePOPs 0.2.6

License: GNU/GPL

Available at: http://www.freepops.org

Homepage: http://www.freepops.org/download.php?module=monitor.lua

Author: Enrico Tassi <gareuselesinge@users.sourceforge.net>

Domain: @monitor

Domain(regex):

Description: Monitors the internal state and statistics of freepops

Parameter:

command one of: stats

6.16 netscape.lua

Name: netscape.net

Version: 0.0.3c

Requires: FreePOPs 0.2.0

License: GNU/GPL

Available at: http://freepops.sourceforge.net/download.php?module=netscape.lua

Homepage: http://freepops.sourceforge.net/

Author: Russell Schwager <russells (at) despammed (.) com>

Domain: @netscape.net

Domain(regex):

Description: This plugin lets you download mail from @netscape.net mailboxes. To use this plugin you have to use your full email address as the username and your real password as the password.

6.17 orange.lua

Name: Orange (ex Wanadoo)

Version: 0.0.8g

Requires: FreePOPs 0.0.99

License: GNU/GPL

Available at: http://www.freepops.org/download.php?contrib=orange.lua

Homepage: http://www.freepops.org/en/viewplugins.php

Authors: TheMarco <themarco (at) fsmail (.) net>, Ernst Vaarties <evaarties (at) xs4all (.) nl>

Domains: @fsmail.net, @wanadoo.nl, @orange.nl, @bedrijfsnaam.nl

Domain(regex):

Description: Orange (ex Wanadoo) webmail plugin. Use your full email address as the username and your real password as the password. For support, please post your questions to the forum.

Parameters:

folder The folder to interact with. Default is inbox, other values are: junk, sent, trash, draft or user defined folder.

emptytrash Not recommended. Forces the plugin to empty the trash when it is done pulling messages. Set it to 1 to activate it.

6.18 popforward.lua

Name: POPforward

Version: 0.0.5

Requires: FreePOPs 0.2.7

License: GNU/GPL

Available at: http://www.freepops.org/download.php?module=popforward.lua

Homepage: http://www.freepops.org/

Author: Enrico Tassi <gareuselesinge (at) users (.) sourceforge (.) net>

Domain: @...

Domain(regex):

Description: This is a POP3 proxy. Read the parameters to know the additional features

Parameters:

realusername If you start the plugin with the username foo@popforward.lua then you need this option to set the real username

host The POP3 server hostname. You can specify the port in the hostname:portnumber way. If host is a lua function (you can set that changing the config.lua file but not on the fly) it is called with the username and should return both host and port.

port To choose to server port to connect to. Use this if you have not specifyed the port in the host parameter.

pipe Pipe the messages to the specified command before passing them to the mail client. Example: '/usr/bin/spamc -t 10'

pipe_limit Limit the maximum size of piped messages, the default (0) is to pipe all of them.

6.19 softhome.lua

Name: softhome.net

Version: 0.0.2b

Requires: FreePOPs 0.0.97

License: GNU/GPL

Available at: http://freepops.sourceforge.net/download.php?contrib=softhome.lua

Homepage: http://freepops.sourceforge.net/

Author: Russell Schwager <russells (at) despammed (.) com>

Domain: @softhome.net

Domain(regex):

Description: To use this plugin you have to use your full email address as the username and your real password as the password. For support, please post a question to the forum instead of emailing the author(s).

Parameter:

folder The folder you want to interact with. Default is Inbox.

6.20 squirrelmail.lua

Name: SquirrelMail

Version: 0.0.4

Requires: FreePOPs 0.2.0

License: GNU/GPL

Available at: http://www.freepops.org/download.php?module=squirrelmail.lua

Homepage: http://www.freepops.org/

Author: Eddi De Pieri <dpeddi (at) users (.) sourceforge (.) net>

Domain: @...

Domain(regex):

Description: This plugin supports webmails made with squirrelmail. You have to hack by hand the plugin to make it work with your website. since now it supports only version 1.2.

6.21 supereva.lua

Name: Supereva

Version: 0.2.7b

Requires: FreePOPs 0.2.0

License: GNU/GPL

Available at: http://www.freepops.org/download.php?module=supereva.lua

Homepage: http://www.freepops.org

Authors: Andrea Dalle Molle <Tund3r (at) fastwebnet (dot) it>, Enrico Tassi <gareuselesinge (at) users (dot) sourceforge (dot) net>, Visioning <unknown>, Viruzzo <unknown>

Domains: @supereva.it, @supereva.com, @freemail.it, @freeweb.org, @mybox.it, @superdada.com, @cicciociccio.com, @mp4.it, @dadacasa.com, @clarence.com, @concento.it, @dada.net

Domain(regex):

Description: This plugin is for the supereva.it portal

6.22 tin.lua

Name: Tin.IT

Version: 0.2.11h

Requires: FreePOPs 0.2.0

License: GNU/GPL

Available at: http://www.freepops.org/download.php?module=tin.lua

Homepage: http://www.freepops.org/

Author: Enrico Tassi <gareuselesinge (at) users (.) sourceforge (.) net>

Domains: @tin.it, @virgilio.it, @alice.it, @tim.it, @atlantide.it

Domain(regex):

Description: This plugin is for italian users only.

Parameters:

folder

limit

6.23 tre.lua

Name: Tre

Version: 0.0.5

Requires: FreePOPs 0.2.0

License: GNU/GPL

Available at: http://www.freepops.org/download.php?module=tre.lua

Homepage: http://www.freepops.org/

Author: Eddi De Pieri <dpeddi (at) users (.) sourceforge (.) net>

Domains: @tre.it, @three.com.au

Domain(regex):

Description: To use this plugin you have to configure your mail client using as username your phone number formatted as 393921234567@three.com.XX and as password the usim's original pin code provided by three. PS: this plugin is tested only whith "three italy", please report me if it works with the three webmail of your country!.

Parameters:

purge Remove automatically deleted mail from trashcan. Permitted flags: yes/no Ie: 443921234567@three.com.au?purge=yes

folder Select a folder (inbox is the default folder). Standard folders are INBOX, INBOX.Draft, INBOX.Sent, INBOX.trash. If you have creaded some your own folders you can select it using its name with the "INBOX." suffix. ie: 443921234567@three.com.au?folder=INBOX.Example

6.24 updater.lua

Name: updater

Version: 0.2.4

Requires: FreePOPs 0.2.3

License: GNU/GPL

Available at: http://freepops.sourceforge.net/download.php?module=updater.lua

Homepage: http://www.freepops.org/

Author: Russell Schwager <russells (at) despammed (.) com>

Domain: @updater

Domain(regex):

Description: This plugin is used to retrieve updated lua modules for freepops. To use this plugin correctly, you need to set the settings for the account created for this to 'leave mail on server'. The first time, you use this account, all the plugins will be retrieved.

Parameter:

modlist The list of modules to update, separated by ','. Example: aaa@updater?modli

6.25 yahoo.lua

Name: yahoo.com

Version: 0.2.20081116

Requires: FreePOPs 0.2.8

License: GNU/GPL

Available at: http://www.freepops.org/download.php?module=yahoo.lua

Homepage: http://www.freepops.org/

Authors: Russell Schwager <russell822 (at) yahoo (.) com>, Nicola Cocchiaro <ncocchiaro (at) users (.) sourceforge (.) net>

Domains: @yahoo.com, @yahoo.ie, @yahoo.it, @yahoo.ca, @rocketmail.com, @yahoo.com.ar, @yahoo.co.in, @yahoo.co.id, @yahoo.com.tw, @yahoo.co.uk, @yahoo.com.cn, @yahoo.es, @yahoo.de, @talk21.com, @btinternet.com, @yahoo.com.au, @yahoo.co.nz, @ymail.com, @yahoo.in

Domain(regex):

Description: This is the webmail support for @yahoo.com, @yahoo.ca and @yahoo.it and similar mailboxes. To use this plugin you have to use your full email address as the user name and your real password as the password.

Parameters:

folder Parameter is used to select the folder (Inbox is the default) that you wish to access. The folders that are available are the standard Yahoo folders, called Inbox, Draft, Sent, Bulk and Trash (for yahoo.it domains you may use the same folder names or the corresponding names in Italian: InArrivo, Bozza, Inviati,Anti-spam, Cestino). For user defined folders, use their name as the value.

view Parameter is used when getting the list of messages to pull. It determines what messages to be pulled. Possible values are All, Unread and Flag.

markunread Parameter is used to have the plugin mark all messages that it pulls as unread. If the value is 1, the behavior is turned on.

nossl Parameter is used to force the module to login through plain HTTP and not HTTPS with SSL. If the value is 1, the SSL is not used.

emptytrash Parameter is used to force the plugin to empty the trash folder when it is done pulling messages. Set the value to 1.

emptybulk Parameter is used to force the plugin to empty the bulk folder when it is done pulling messages. Set the value to 1.

maxmsgs Parameter is used to force the plugin to only download a maximum number of messages.

keepmsgstatus Parameter is used to maintain the status of the message in the state it was before being pulling. If the value is 1, the behavior is turned on and will override the markunread flag.

useeventlib Parameter is used to turn on windows event library logging. If the value is 1, the behavior is turned on.

domain Parameter is used to override the domain in the email address. This is used so that users don't need to add a mapping to config.lua for a hosted hotmail account.

6.26 aggregator.lua

Name: RSS/RDF aggregator

Version: 0.2.8

Requires: FreePOPs 0.2.0

License: GNU/GPL

Available at: http://www.freepops.org/download.php?module=aggregator.lua

Homepage: http://www.freepops.org/

Authors: Simone Vellei <simone_vellei (at) users (.) sourceforge (.) net>, Fernando Lucas Rodriguez <info (at) fernandolucas (.) es>

Domains: @aggregator, @...

Domain(regex):

Description: Usually you can benefit from the W3C's RSS for mat when you read some website news. The RSS file indexes the news, providing a link to them. This plugin can make your mail client see the RSS file as a mailbox from which you can download each news as if it was a mail message. The only limitation is that this plugin can fetch only a news summary plus the news link. To use this plugin you have to use a casual user name with the @aggregator suffix (ex: foo@aggregator) and as the password the URL of the RSS file(ex: http://www.securityfocus.com/rss/vulnerabilities.xml). For your commodity we added some alias for you. This means you have not to search by hand the URL of the RSS file. We added some domain, for example @securityfocus.com, that can be used to directly use the aggregator plugin with these website. To use these alias you have to use a user name in the form something@aggregatordomain and a casual password.

This is the list of aliases for the aggregator plugin.

aggregatordomain	description
freepops.rss.en	http://www.freepops.org/ news (English)
freepops.rss.it	http://www.freepops.org/ news (Italian)
flatnuke.sf.net	http://flatnuke.sourceforge.net/ news (Italian)
ziobudda.net	http://ziobudda.net/ news (both Italian and English)
punto-informatico.it	http://punto-informatico.it/ news (Italian)
linuxdevices.com	http://linuxdevices.com/ news (English)
gaim.sf.net	http://gaim.sourceforge.net/ news (English)
securityfocus.com	http://www.securityfocus.com/ new vulnerabilities (English)
games.gamespot.com	http://www.gamespot.com/ computer games news (English)
news.gamespot.com	http://www.gamespot.com/ GameSpot news (English)
kerneltrap.org	http://kerneltrap.org news (English)
mozillaitalia.org	http://www.mozillaitalia.org news (Italian)
linux.kerneltrap.org	http://linux.kerneltrap.org news (English)
linuxgazette.net	http://linuxgazette.net news (English)

6.27 flatnuke.lua

Name: flatnuke

Version: 0.0.6

Requires: FreePOPs 0.2.0

License: GNU/GPL

Available at: http://www.freepops.org/download.php?module=kernel.lua

Homepage: http://www.freepops.org/

Author: Simone Vellei <simone_vellei (at) users (.) sourceforge (.) net>

Domains: @flatnuke, @...

Domain(regex):

Description: This plugin is an aggregator plugin specialized in the websites made with the FlatNuke content management system, or other sites that use the same news format like the FreePOPs website. Since in a FlatNuke site the news are stored in plain XML files this plugin is able to fetch the whole news, and not only the headings as the aggregator plugin does. This is really useful if you don't want to surf the website to read the news. To use this plugin you have to use a user name with the @flatnuke domain (ex: something@flatnuke) and a flatnuke homepage URL as the password (ex: http://flatnuke.sourceforge.net/, no need for the RSS file URL since FlatNuke puts the RSS in a fixed and well known position.

There are some alias for FlatNuke sites, see the aggregator plugin documentation to know what this means:

aggregatordomain	description
freepops.en	http://www.freepops.org/ full news (English)
freepops.it	http://www.freepops.org/ full news (Italian)
flatnuke.it	http://flatnuke.sourceforge.net/ full news (Italian)

7 Creating a plugin

Two sections follow, the first is a quick overview of what a plugin has to do, the latter is a more detailed tutorial. Before proceeding I suggest you read some stuff that is at the base of plugin writing:

1. Since plugins are written in LUA you must read at least the LUA tutorial (HTTP://lua-users.org/wiki/LuaTutorial); many thanks to the guys who wrote it. LUA is a quite simple scripting language, easy to learn, and easy to read. If you are interested in this language you should read THE book about LUA ("Programming in Lua" by Roberto Ierusalimschy HTTP://www.inf.puc-rio.br/~roberto/book/). It is a really good book, believe me. Today I've seen that the book is completely available online here. HTTP://www.lua.org/pil/

2. Since we have to implement a POP3 backend you should know what POP3 is. The RFC is number 1939 and is included in the doc/ directory of the source package of FreePOPs, but you can fetch it from the net HTTP://www.ietf.org/rfc/rfc1939.txt.

3. Read carefully this tutorial, it is hardly a good tutorial, but is better than nothing.

4. The website contains, in the doc section, a quite good documentation of the sources. You should keep a web browser open at the LUA modules documentation page while writing a plugin.

5. After creating a prototype, you should read a full featured plugin. The libero.lua plugin is really well commented, you may start there.

6. Remember that this software has an official forum (HTTP://freepops.diludovico.it) and some authors you may ask for help.

7. FreePOPs is licensed under the GNU/GPL license, therefore any and all software that makes use of its code must be released under the same license. This includes plugins. For more information read the enclosed COPYING file or see the license text at HTTP://www.gnu.org/licenses/gpl.html.

7.1 Plugins overview

A plugin is essentially a backend for a POP3 server. The plugins are written in LUA[1] while the POP3 server is written in C. Here we examine the interfaces between The C core and the LUA plugins.

7.1.1 The interface between the C core and a plugin

As we explained before the C POP3 frontend has to be attached to a LUA backend. The interface is really simple if you know the POP3 protocol. Here we only summarize the meaning, but the RFC 1939 (included in the `doc/` directory of the source distribution) is really short and easy to read. As your intuition should suggest the POP3 client may ask the pop3 server to know something about the mail that is in the mailbox and eventually retrieve/delete a message. And this is exactly what it does.

The backend must implement all the POP3 commands (like USER, PASS, RETR, DELE, QUIT, LIST, ...) and must give back to the frontend the result. Let us give a simple example of a POP3 session taken from the RFC:

```
 1  S: <wait for connection on TCP port 110>
 2  C: <open connection>
 3  S:    +OK POP3 server
 4  C:    USER linux@kernel.org
 5  S:    +OK now insert the password
 6  C:    PASS gpl
 7  S:    +OK linux's maildrop has 2 messages (320 octets)
 8  C:    STAT
 9  S:    +OK 1 320
10  C:    LIST
11  S:    +OK 2 messages (320 octets)
12  S:    1 320
13  S:    .
14  C:    RETR 1
15  S:    +OK 120 octets
16  S:    <the POP3 server sends message 1>
17  S:    .
18  C:    DELE 1
19  S:    +OK message 1 deleted
20  C:    QUIT
21  S:    +OK dewey POP3 server signing off (maildrop empty)
22  C:  <close connection>
23  S:  <wait for next connection>
```

[1] The language website is HTTP://www.lua.org

In this session the backend will be called for lines 4, 6, 8, 10, 14, 18, 20 (all the `C:` lines) and respectively the functions implementing the POP3 commands will be called this way

```
user(p,"linux@kernel.org")
pass(p,"gpl")
stat(p)
list_all(p)
retr(p,1)
dele(p,1)
quit_update(p)
```

Later I will make clear what p is. I hope we'll remove it making it implicit for complete transparency. It is easy to understand that there is a 1-1 mapping between POP3 commands and plugin function calls. You can view a plugin as the implementation of the POP3 interface.

7.1.2 The interface between a plugin and the C core

Let us take in exam the call to `pass(p,"gpl")`. Here the plugin should authenticate the user (if there is a sort of authentication) and inform the C core of the result. To achieve this each plugin function must return an error flag, to be more accurate one of these errors:

Code	Meaning
POPSERVER_ERR_OK	No error
POPSERVER_ERR_NETWORK	An error concerning the network
POPSERVER_ERR_AUTH	Authorization failed
POPSERVER_ERR_INTERNAL	Internal error, please report the bug
POPSERVER_ERR_NOMSG	The message number is out of range
POPSERVER_ERR_LOCKED	Mailbox is locked by another session
POPSERVER_ERR_EOF	End of transmission, used in the popserver_callback
POPSERVER_ERR_TOOFAST	You are not allowed to reconnect to the server now, wait a bit and retry
POPSERVER_ERR_UNKNOWN	No idea of what error I've encountered

In our case the most appropriate error codes are `POPSERVER_ERR_AUTH` and `POPSERVER_ERR_OK`. This is a simple case, in which an error code is enough. Now we analyze the more complex case of the call to `list_all(p)`. Here we have to return an error code as before, but we also have to inform the C core of the size of all messages in the mailbox. We need the p parameter passed to each plugin function (note that that parameter may became implicit in the future). `p` stands for the data structure that the C core expects us to fill calling

appropriate functions like `set_mailmessage_size(p,num,size)` where num is the message number and size is the size in bytes. Usually it is really common to put more functions all together. For example when you get the message list page in a webmail you know the number of the messages, their size and uidl so you can fill the p data structure with all the informations for LIST, STAT, UIDL.

The last case that we examine is `retr(p,num,data)`. Since a mail message can be really big, there is no pretty way of downloading the entire message without making the mail client complain about the server death. The solution is to use a callback. Whenever the plugin has some data to send to the client he should call the `popserver_callback(buffer,data)`. `data` is an opaque structure the popserver needs to accomplish its work (note that this parameter may be removed for simplicity). In some cases, for example if you know the message is small or you are working on a fast network, you can fetch the whole message and send it, but remember that this is more memory consuming.

7.2 The art of writing a plugin (plugins tutorial)

In this section we will write a plugin step by step, examining each important detail. We will not write a real and complete plugin since it may be a bit hard to follow but we will create an ad-hoc webmail for our purposes.

7.2.1 (step 1) The skeleton

The first thing we will do is copy the `skeleton.lua` file to `foo.lua` (since we will write the plugin for the *foo.xx* webmail, *xx* stands for a real domain, but I don't want to mention any websites here...). Now with your best editor (I suggest vim under Unix and scintilla for win32, since they support syntax highlights for LUA, but any other text editor is OK) open `foo.lua` and change the first few lines adding the plugin name, version, your name, your email and a short comment in the proper places.

```
-- ************************************************************************** --
--  FreePOPs @--put here domain-- webmail interface
--
--  $Id: manual.tex,v 1.41 2007/10/28 12:32:22 gareuselesinge Exp $
--
--  Released under the GNU/GPL license
--  Written by --put Name here-- <--put email here-->
-- ************************************************************************** --

PLUGIN_VERSION = "--put version here--"
PLUGIN_NAME = "--put name here--"
```

Now we have an empty plugin, but it is not enough to start hacking on it. We need to open the `config.lua` file (in the win32 distribution it is placed in the

main directory, while in the Unix distribution it is in `/etc/freepops/`; other copies of this file may be included in the distributions, but they are backup copies) and add a line like this

```
-- foo plugin
freepops.MODULES_MAP["foo.xx"]      = {name="foo.lua"}
```

at the beginning of the file. Before ending the first step you should try if the plugin is correctly activated by FreePOPs when needed. To do this we have to add few lines to `foo.lua`, in particular we have to add an error return value to `user()`.

```
-- ---------------------------------------------------------------------- --
-- Must save the mailbox name
function user(pstate,username)
        return POPSERVER_ERR_AUTH
end
```

Now the user function always fails, returning an authentication error. Now you have to start FreePOPs (if it is already running you don't have to restart it) and start telnet (under win32 you should open a DOS prompt, under Unix you have the shell) and type `telnet localhost 2000` and then type `user test@foo.xx`.

```
tassi@garfield:~$ telnet localhost 2000
Trying 127.0.0.1...
Connected to garfield.
Escape character is '^]'.
+OK FreePOPs/0.0.10 pop3 server ready
user test@foo.xx
-ERR AUTH FAILED
Connection closed by foreign host.
```

The server responds closing the connection and printing an authorization failed message (thats OK, since the `user()` function of our plugin returns this error). In the standard error file (the console under Unix, the file `stderr.txt` under Windows) the error messages get printed, don't mind them now.

7.2.2 (step 2) The login

The login procedure is the first thing we have to do. The POP3 protocol has 2 commands for login, *user* and *pass*. First the client does a user, then it tells the server the password. As we have already seen in the overview this means that first `user()` and then `pass()` will be called. This is a sample login:

```
tassi@garfield:~$ telnet localhost 2000
Trying 127.0.0.1...
Connected to garfield.
Escape character is '^]'.
+OK FreePOPs/0.0.10 pop3 server ready
user test@foo.xx
+OK PLEASE ENTER PASSWORD
pass hello
-ERR AUTH FAILED
```

If you start FreePOPs with the `-w` switch you should read this on standard error/standard output:

```
freepops started with loglevel 2 on a little endian machine.
Cannot create pid file "/var/run/freepopsd.pid"
DBG(popserver.c, 162): [5118] ?? Ip address 0.0.0.0 real port 2000
DBG(popserver.c, 162): [5118] ?? Ip address 127.0.0.1 real port 2000
DBG(popserver.c, 162): [5118] -> +OK FreePOPs/0.0.10 pop3 server ready
DBG(popserver.c, 162): [5118] <- user test@foo.xx
DBG(log_lua.c,  83): (@src/lua/foo.lua, 37) : FreePOPs plugin 'Foo web mail' version '0.
*** the user wants to login as 'test@foo.xx'
DBG(popserver.c, 162): [5118] -> +OK PLEASE ENTER PASSWORD
DBG(popserver.c, 157): [5118] <- PASS *********
*** the user inserted 'hello' as the password for 'test@foo.xx'
DBG(popserver.c, 162): [5118] -> -ERR AUTH FAILED
AUTH FAILED
DBG(threads.c,  81): thread 0 will die
```

and the plugin has been changed a bit to store the user login and print some debug info. This is the plugin that gave this output:

```
foo_globals= {
       username="nothing",
       password="nothing"
}
-- ----------------------------------------------------------------------- --
-- Must save the mailbox name
function user(pstate,username)
        foo_globals.username = username
        print("*** the user wants to login as '"..username.."'")
        return POPSERVER_ERR_OK
end
-- ----------------------------------------------------------------------- --
-- Must login
function pass(pstate,password)
```

```
        foo_globals.password = password
        print("*** the user inserted '"..password..
            "' as the password for '"..foo_globals.username.."'")
        return POPSERVER_ERR_AUTH end
-- ------------------------------------------------------------------------ --
-- Must quit without updating
function quit(pstate)
        return POPSERVER_ERR_OK
end
```

Here we have some important news. First the `foo_globals` table that will contain all the globals (values that should be available to successive function calls) we need. So far we have the username and password there. The `user()` function now stores the passed username in the `foo_globals` table and prints something on standard output. The `pass()` function likewise stores the password in the global table and prints some stuff. The `quit()` function simply returns `POPSERVER_ERR_OK` to make FreePOPs happy.

Now that we know how FreePOPs will act during the login we have to implement the login in the webmail, but first uncomment the few lines in the `init()` function (that is called when the plugin is started), that loads the `browser.lua` module (the module we will use to login in the webmail). Here is the webmail login page viewed with Mozilla and the source of the page (you can see it with Mozilla with Ctrl-U, figure 5).



Figure 5: login

```
<html>
<head>
<title>foo.xx webmail login</title>
</head>
<body style="background-color : grey; color : white">
<h1>Webmail login</h1>
<form name="webmail" method="post" action="http://localhost:3000/">
```

```
login: <input type="text" size="10" name="username"> <br>
password: <input type="password" size="10" name="password"> <br>
<input type="submit" value="login">
</form>
</body>
</html>
```

Here we have 2 input fields, one called username and one called password. When the user clicks login the web browser will `POST` to `HTTP://localhost:3000/` the form contents (I used a local address for comfort, but it should be something like `HTTP://webmail.foo.xx/login.php`). This is what the browser sends:

```
POST / HTTP/1.1
Host: localhost:3000
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.6) Gecko/20040614 Firefox/0.8 A
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 37

username=test%40foo.xx&password=hello
```

We are not interested in the first part (the HTTP header, since the browser module will take care of it) but in the last part, the posted data. Since the fields of the form were username and password, the posted data is `username=test%40foo.xx&password=hello`. Now we want to reproduce the same HTTP request with our plugin. This is the simple code that will do just that.

```
-- ------------------------------------------------------------------------ --
-- Must login
function pass(pstate,password)
       foo_globals.password = password

       print("*** the user inserted '"..password..
            "' as the password for '"..foo_globals.username.."'")

       -- create a new browser
       local b = browser.new()

       -- store the browser object in globals
```

```
        foo_globals.browser = b

        -- create the data to post
        local post_data = string.format("username=%s&password=%s",
                foo_globals.username,foo_globals.password)
        -- the uri to post to
        local post_uri = "http://localhost:3000/"

        -- post it
        local file,err = nil, nil

        file,err = b:post_uri(post_uri,post_data)

        print("we received this webpage: ".. file)
        return POPSERVER_ERR_AUTH
end
```

First we create a browser object, then we build the `post_uri` and `post_data` using a simple `string.format` (printf-like function). And this is the resulting request

```
POST / HTTP/1.1
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.6) Gecko/20040322 Firefox/0.8
Pragma: no-cache
Accept: */*
Host: localhost
Content-Length: 35
Content-Type: application/x-www-form-urlencoded

username=test@foo.xx&password=hello
```

that is essentially the same (we should url-encode the post data with `curl.escape()`) we wanted to do. We saved the browser object to the global table, since we want to use the same browser all the time.

Now that we have logged in, we want to check the resulting page, and maybe extract a session ID that will be used later. This is the code to extract the session id and the HTML page we have received in response to the login request

```
        ... the same as before here ...

        print("we received this webpage: ".. file)

        -- search the session ID
        local _,_,id = string.find(file,"session_id=(%w+)")
```

```
        if id == nil then
               return POPSERVER_ERR_AUTH
        end

        foo_globals.session_id = id
        return POPSERVER_ERR_OK
end
```

and figure 6 is the returned web page.



Figure 6: logindone

```
<html>
<head>
<title>foo.xx webmail</title>
</head>
<body style="background-color : grey; color : white">
<h1>Webmail - test@foo.xx</h1>
Login done! click here to view the inbox folder.
<a href="http://localhost:3000/inbox.php?session_id=ABCD1234">inbox</a>
</body>
</html>
```

Note that we extracted the session ID string using `string.find(file,"session_id=(%w+)")`. This is a really important function in the lua library and, even if it is described in the lua tutorial at HTTP://lua-users.org, we will talk a bit about captures here. Look at the page source. We are interested in the line `<a href="HTTP://localhost:3000/inbox.php?session_id=ABCD1234">inbox</a>` that contains the session_id we want to capture. Our expression is *`session_id=(%w+)`* that means we want to match all the strings that start with `session_id=` and than continue with one or more alphanumerical character. Since we wrote `%w+` in round brackets, we mean to capture the content of brackets (the alphanumerical part). So string.find will return 3 values, the first two are ignored (assigned to the dummy variable _) while the third is the captured string

(in our case `ABCD1234`). The LUA tutorial at lua-users is quite good and at HTTP://sf.net/projects/lua-users you can find the LUA short reference that is a summary of all standard lua functions and is a really good piece of paper (so many thanks to Enrico Colombini). If you really like LUA you should buy THE book about LUA called *"Programming in Lua"* by Roberto Ierusalimschy (consider it the K&R for LUA).

7.2.3 (step 3) Getting the list of messages

Now we have to implement the `stat()` function. The stat is probably the most important function. It must retrieve the list of messages in the webmail and their UIDL and size. In our example we will use the mlex module to grab the important info from the page, but you can use the string LUA module to do the same with captures. This is our inbox page (see figure 7)

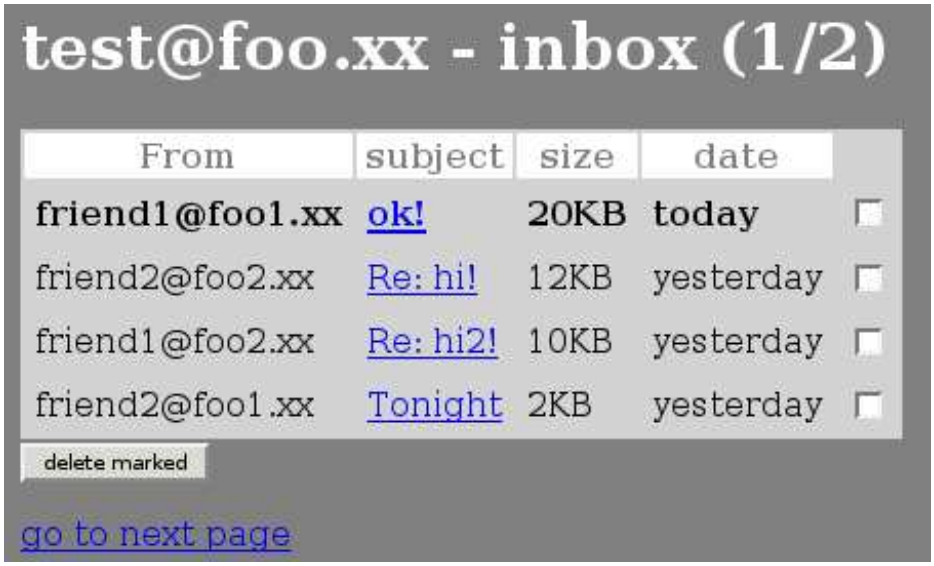


Figure 7: inbox

and this is the HTML body (only the first 2 messages are reported)

```
<h1>test@foo.xx - inbox (1/2)</h1>
<form name="inbox" method="post" action="/delete.php">
<input type="hidden" name="session_id" value="ABCD1234">
<table>
<tr><th>From</th><th>subject</th><th>size</th><th>date</th></tr>
<tr>
  <td><b>friend1@foo1.xx</b></td>
  <td><b><a href="/read.php?session_id=ABCD1234&uidl=123">ok!</a></b></td>
```

```
  <td><b>20KB</b></td>
  <td><b>today</b></td>
  <td><input type="checkbox" name="check_123"></td>
</tr>
<tr>
  <td>friend2@foo2.xx</td>
  <td><a href="/read.php?session_id=ABCD1234&uidl=124">Re: hi!</a></td>
  <td>12KB</td>
  <td>yesterday</td>
  <td><input type="checkbox" name="check_124"></td>
</tr>
</table>
<input type="submit" value="delete marked">
</form>
<a href="/inbox.php?session_id=ABCD1234&page=2">go to next page</a>
</body>
```

We have retrieved the HTML using the browser and the `get_uri()` method (remember the URI for the inbox was in the login page). As you can see the messages are in a table and this table has the same structure for each message. This is the place in which you may use mlex. Just take all the stuff between `<tr>` and `</tr>` of a message row and delete all but the tags name. Then replace every empty space (we call space the string between two tags) with a "`.*`". This is what we have obtained (it should be all in the same line, here is wrapped for lack of space) from the first message.

```
.*<tr>.*<td>.*<b>.*</b>.*</td>.*<td>.*<b>.*<a>.*</a>.*</b>.*</td>.*
<td>.*<b>.*</b>.*</td>.*<td>.*<b>.*</b>.*</td>.*
<td>.*<input>.*</td>.*</tr>
```

This expression is used to match the table row containing info about the message. Now cut and paste the line and replace every space and every tag with O (the letter, not the digit 0) or X. Put an X in the interesting fields (in our example the size and the input tag, that contains the message uidl).

```
0<0>0<0>0<0>0<0>0<0>0<0>0<0>0<0>0<0>0<0>0<0>0<0>0
<0>0<0>X<0>0<0>0<0>0<0>0<0>0<0>0
<0>0<X>0<0>0<0>
```

While the first expression will be used to match the table row, this one will be used to extract the important fields. This is the code that starts mlex on the HTML and fills the popstate data structure with the captured data.

```
-- ------------------------------------------------------------------------ --
-- Fill the number of messages and their size
```

```
function stat(pstate)
	local file,err = nil, nil
	local b = foo_globals.browser
	file,err = b:get_uri("http://localhost:3000/inbox.php?session_id=".. 
		foo_globals.session_id)
	local e = ".*<tr>.*<td>.*<b>.*</b>.*</td>.*<td>.*<b>.*<a>".. 
		".*</a>.*</b>.*</td>.*<td>.*<b>.*</b>.*</td>.*<td>.*".. 
		"<b>.*</b>.*</td>.*<td>.*<input>.*</td>.*</tr>"
	local g = "O<O>O<O>O<O>O<O>O<O>O<O>O<O>O<O>O<O>O<O>O<O>O<O>O<O>O<O>O<O>O".. 
		"<O>O<O>X<O>O<O>O<O>O<O>O<O>O<O>O<O>O<O>O<X>O<O>O<O>"
	local x = mlex.match(file,e,g)
	--debug print
	x:print()

	set_popstate_nummesg(pstate,x:count())
	for i=1,x:count() do
		local _,_,size = string.find(x:get(0,i-1),"(%d+)")
		local _,_,size_mult_k = string.find(x:get(0,i-1),"([Kk][Bb])")
	local _,_,size_mult_m = string.find(x:get(0,i-1),"([Mm][Bb])")
		local _,_,uidl = string.find(x:get(1,i-1),"check_(%d+)")

		if size_mult_k ~= nil then
			size = size * 1024
		end
		if size_mult_m ~= nil then
			size = size * 1024 * 1024
		end

		set_mailmessage_size(pstate,i,size)
		set_mailmessage_uidl(pstate,i,uidl)
	end

	return POPSERVER_ERR_OK
end
```

The result of `x:print()` is the following

```
{'20KB','input type="checkbox" name="check_123"'}
```

and the telnet session follows

```
+OK FreePOPs/0.0.11 pop3 server ready
user test@foo.xx
+OK PLEASE ENTER PASSWORD
pass secret
+OK ACCESS ALLOWED
```

```
stat
+OK 1 20480
quit
+OK BYE BYE, UPDATING
```

We have not listed here how we added the dummy `return POPSERVER_ERR_OK` line to the `quit()` function. The source code listed before uses mlex to extract the two interesting strings, then parses them searching for the size and the size multiplier and the uidl. Then sets the mail message attributes. But here you can see that we just matched the first message. To match the other messages we have to inform the mlex module that the <b> tag is optional (you can see that only the first message is in bold). So we change the expressions to

```
.*<tr>.*<td>[.*]{b}.*{/b}[.*]</td>.*<td>[.*]{b}.*<a>.*</a>.*{/b}[.*]</td>.*
<td>[.*]{b}.*{/b}[.*]</td>.*<td>[.*]{b}.*{/b}[.*]</td>.*
<td>.*<input>.*</td>.*</tr>
```

and

```
O<O>O<O>[O]{O}O{O}[O]<O>O<O>[O]{O}O<O>O<O>O{O}[O]<O>O
<O>[O]{O}X{O}[O]<O>O<O>[O]{O}O{O}[O]<O>O
<O>O<X>O<O>O<O>
```

Now the stat command responds with `+OK 4 45056` and the debug print is

```
{'20KB','input type="checkbox" name="check_123"'}
{'12KB','input type="checkbox" name="check_124"'}
{'10KB','input type="checkbox" name="check_125"'}
{'2KB','input type="checkbox" name="check_126"'}
```

Now we have a proper function stat that fill the popstate data structure with the info the popserver needs to respond to a stat request. Since the list, uidl, list_all and uidl_all requests can be satisfied with the same data we will use the standard function provided by the common.lua module. It will be explained in the next step, but we have to add 2 important lines to the `stat()` function, to avoid a double call.

```
function stat(pstate)
       if foo_globals.stat_done == true then return POPSERVER_ERR_OK end

       ... the same code here ...

       foo_globals.stat_done = true
       return POPSERVER_ERR_OK
end
```

The most important function is done, but a lot of notes must be written here. First, mlex is really comfortable sometimes, but you may find more helpful using the lua string library or the regularexp library (posix extended regular expressions) to reach the same point. Second, this implementation stops at the first inbox page. You should visit all the inbox pages maybe using the `do_until()` function in the `support.lua` library (that will be briefly described at the end of this tutorial). Third we make no error checking. For example the file variable may be nil and we must check these things to make a good plugin.

7.2.4 (step 4) The common functions

The common module gives us some pre-cooked functions that depend only on a well implemented `stat()` (I mean a stat than can be called more than once). This is our implementation of these functions

```
-- ------------------------------------------------------------------------ --
-- Fill msg uidl field
function uidl(pstate,msg) return common.uidl(pstate,msg) end

-- ------------------------------------------------------------------------ --
-- Fill all messages uidl field
function uidl_all(pstate) return common.uidl_all(pstate) end

-- ------------------------------------------------------------------------ --
-- Fill msg size
function list(pstate,msg) return common.list(pstate,msg) end

-- ------------------------------------------------------------------------ --
-- Fill all messages size
function list_all(pstate) return common.list_all(pstate) end

-- ------------------------------------------------------------------------ --
-- Unflag each message marked for deletion
function rset(pstate) return common.rset(pstate) end

-- ------------------------------------------------------------------------ --
-- Mark msg for deletion
function dele(pstate,msg) return common.dele(pstate,msg) end

-- ------------------------------------------------------------------------ --
-- Do nothing
function noop(pstate) return common.noop(pstate) end
```

but first add the common module loading code to your `init()` function

```
        ... the same code ..

        -- the common module
        require("common")

        -- checks on globals
        freepops.set_sanity_checks()

        return POPSERVER_ERR_OK
end
```

7.2.5 (step 5) Deleting messages

Deleting messages is usually a normal post and an example of the `post_data` is `session_id=ABCD1234&check_124=on&check_126=on`. The code follows

```
-- ----------------------------------------------------------------------- --
-- Update the mailbox status and quit
function quit_update(pstate)
      -- we need the stat
      local st = stat(pstate)
      if st ~= POPSERVER_ERR_OK then return st end

      -- shorten names, not really important
      local b = foo_globals.b
      local post_uri = b:wherearewe() .. "/delete.php"
      local session_id = foo_globals.session_id
      local post_data = "session_id=" .. session_id .. "&"

      -- here we need the stat, we build the uri and we check if we
      -- need to delete something

      local delete_something = false;
      for i=1,get_popstate_nummesg(pstate) do
     if get_mailmessage_flag(pstate,i,MAILMESSAGE_DELETE) then
      post_data = post_data .. "check_" ..
     get_mailmessage_uidl(pstate,i).. "=on&"
delete_something = true
             end
      end

      if delete_something then
             b:post_uri(post_uri,post_data)
      end

      return POPSERVER_ERR_OK
end
```

Consider we do the post only if at least one message is marked for deletion. Another important think to keep in mind is that making only one post for all messages is better than making a single post for each message. When it is possible you should reduce the number of HTTP requests as much as you can since it is here we move FreePOPs from a rabbit to a tortoise.

7.2.6 (step 6) Downloading messages

You may ask why I talk about this only at point 6, while having the mail is probably what you want from a plugin. Implementing the `retr()` function is usually simple. It really depends on the webmail, but here we will talk of the simple case, while at the end of the tutorial you will see how to deal with complex webmails. The simple case is the one in which the webmail has a save message button. And the saved message is a plain text file containing both the header and the body of the message. There are only two interesting points in this case, firstly big messages, secondly the dot issue.

Big messages are a cause of timeout. Yes, the most simple way of downloading a message is calling `b:get_uri()` and store the message in a variable, and then send it to the mail client with `popserver_callback()`. But think that a 5MB mail, downloaded with a 640Kbps DSL connection, at full 80KBps speed, takes 64 seconds to download. This means your plugin will not send any data to the mail client for more than one minute and this will make the mail client to disconnect from FreePOPs thinking the POP3 server is dead. So we must send the data to the mail client as soon as we can. For this we have the `b:pipe_uri()` function that calls a callback whenever it has some fresh data. The following code is the callback factory function, that creates a new callback to pass to the `pipe_uri` browser method.

```
-----------------------------------------------------------------------------
-- The callback factory for retr
--
function retr_cb(data)
        local a = stringhack.new()
        return function(s,len)
                s = a:dothack(s).."\0"
                popserver_callback(s,data)
                return len,nil
        end
end
```

Here you see that the callback simply uses `popserver_callback()` to pass the data to the mail client, but before doing this it mangles the data with the stringhack. But this is the second interesting point.

The POP3 protocol should end the retr command answer with a line that contains only 3 bytes, "`.\r\n`". But what if a line, inside the mail body, is

a simple point? We have to escape it to "`..\r\n`". This is not so hard, a `string.gsub(s,"\r\n.\r\n","\r\n..\r\n")` is all we need... but not in the case of callbacks. The send callback will be called with some fresh data, and called more than once if the mail is big. And if the searched pattern is truncated between two calls the `string.gsub()` method will fail. This is why the stringhack module helps us. The `a` object lives as long as the callback function will be called (see the closure page of the lua tutorial) and will keep in mind that the searched pattern may be truncated.

Finally the `retr()` code

```
-- ----------------------------------------------------------------------- --
-- Get message msg, must call
-- popserver_callback to send the data
function retr(pstate,msg,pdata)
        -- we need the stat
        local st = stat(pstate)
        if st ~= POPSERVER_ERR_OK then return st end

        -- the callback
        local cb = retr_cb(data)

        -- some local stuff
       local session_id = foo_globals.session_id
       local b = internal_state.b
       local uri = b:wherearewe() .. "/download.php?session_id="..session_id..
                "&message="..get_mailmessage_uidl(pstate,msg)

        -- tell the browser to pipe the uri using cb
        local f,rc = b:pipe_uri(uri,cb)
        if not f then
                log.error_print("Asking for "..uri.."\n")
                log.error_print(rc.."\n")
                return POPSERVER_ERR_NETWORK
        end
end
```

7.2.7 (step 7) Test it

Making a good plugin needs a lot of testing. You should ask for beta testers at the FreePOPs forum (HTTP://freepops.diludovico.it) and ask the software authors to include it in the main distribution. You should also read the webmail contract, check if there is something like "*I'll never use webmail->pop3 server to read my mail*" and send a copy to the authors of the software.

7.2.8 (step 8) The so mentioned last part of the tutorial

There are a lot of things we have omitted here.

The multi-page stat is the real good implementation for `stat()`. We mentioned before that our implementation lists only the messages in the first page. The code for parsing and extracting interesting info from a page is already written, we simply need a function that checks if we are in the last page and if not it changes the value of a `uri` variable. The `uri` variable will be used by the fetch function. In this case you should use the support module with the do_until cycle. This is a simple example of `do_until()`

```
-- ------------------------------------------------------------------------ --
-- Fill the number of messages and their size
function stat(pstate)
        ... some code as before ...

        -- this string will contain the uri to get. it may be updated by
        -- the check_f function, see later
        local uri = string.format(libero_string.first,popserver,session_id)

        -- The action for do_until
        --
        -- uses mlex to extract all the messages uidl and size
        local function action_f (s)
                -- calls match on the page s, with the mlexpressions
                -- statE and statG
                local x = mlex.match(s,e,g)

                -- the number of results
                local n = x:count()

                if n == 0 then return true,nil end

                -- this is not really needed since the structure
                -- grows automatically... maybe... don't remember now
                local nmesg_old = get_popstate_nummesg(pstate)
                local nmesg = nmesg_old + n
                set_popstate_nummesg(pstate,nmesg)

                -- gets all the results and puts them in the popstate structure
                        ... some code as before ...

                        set_mailmessage_size(pstate,i+nmesg_old,size)
                        set_mailmessage_uidl(pstate,i+nmesg_old,uidl)
                end
```

```
        return true,nil
end

-- check must control if we are not in the last page and
-- eventually change uri to tell retrieve_f the next page to retrieve
local function check_f (s)
        local tmp1,tmp2 = string.find(s,next_check)
        if tmp1 ~= nil then
                -- change retrieve behavior
                uri = "--build the uri for the next page--"

                -- continue the loop
                return false
        else
                return true
        end
end

-- this is simple and uri-dependent
local function retrieve_f ()
        local f,err = b:get_uri(uri)
        if f == nil then
                return f,err
        end

        local _,_,c = string.find(f,"--timeout string--")
        if c ~= nil then
                internal_state.login_done = nil
                session.remove(key())
                local rc = libero_login()
                if rc ~= POPSERVER_ERR_OK then
                        return nil,"Session ended,unable to recover"

                uri = "--uri for the first page--"
                return b:get_uri(uri)
         end

         return f,err
end

-- initialize the data structure
set_popstate_nummesg(pstate,0)

-- do it
if not support.do_until(retrieve_f,check_f,action_f) then
          log.error_print("Stat failed\n")
```

```
                session.remove(key())
                return POPSERVER_ERR_UNKNOWN
        end

        -- save the computed values
        internal_state["stat_done"] = true
        return POPSERVER_ERR_OK
end
```

The only strange things are the retrieve function and the session saving stuff. Since webmail sometimes timeout you should check if the retrieved page is valid or not, and eventually retry the login. The session saving is the next issue.

Saving the session is the way to make FreePOPs really similar to a browser. This means the next time you check the mail FreePOPs will simply reload the inbox page and won't login again. To do this you need a `key()` function that gives a unique ID for each session

```
------------------------------------------------------------------------------
-- The key used to store session info
--
-- This key must be unique for all webmails, since the session pool is one
-- for all the webmails
--
function key()
        return foo_globals.username .. foo_globals.password
end
```

and a `foo_globals` serialization function

```
------------------------------------------------------------------------------
-- Serialize the internal state
--
-- serial.serialize is not enough powerful to correctly serialize the
-- internal state. The field b is the problem. b is an object. This means
-- that it is a table (and no problem for this) that has some field that are
-- pointers to functions. this is the problem. there is no easy way for the
-- serial module to know how to serialize this. so we call b:serialize
-- method by hand hacking a bit on names
--
function serialize_state()
        internal_state.stat_done = false;
        return serial.serialize("foo_globals",foo_globals) ..
                internal_state.b:serialize("foo_globals.b")
end
```

Now you have to tell FreePOPs to save the state in the `quit_update()` function and load it back in the `pass()` one. This is the new `pass()` structure

```
function pass(pstate,password)
        -- save the password
        internal_state.password = password

        -- eventually load session
        local s = session.load_lock(key())

        -- check if loaded properly
        if s ~= nil then
                 -- "\a" means locked
                 if s == "\a" then
                          log.say("Session for "..internal_state.name..
                              " is already locked\n")
                          return POPSERVER_ERR_LOCKED
                 end

                 -- load the session
                 local c,err = loadstring(s)
                 if not c then
                          log.error_print("Unable to load saved session: "..err)
                          return foo_login()
                 end

                 -- exec the code loaded from the session string
                 c()

                log.say("Session loaded for " .. internal_state.name .. "@" ..
                         internal_state.domain ..
                         "(" .. internal_state.session_id .. ")\n")

                return POPSERVER_ERR_OK
        else
                -- call the login procedure
                return foo_login()
        end
end
```

where `foo_login()` is the old `pass()` function with minor changes. Don't forget to call `session.unlock(key())` in the `quit()` function, since you have to release the session in case of failure (and `quit()` is called here) and to save the session in `quit_update()`

```
-- save fails if it is already saved
session.save(key(),serialize_state(),session.OVERWRITE)
-- unlock is useless if it have just been saved, but if we save
-- without overwriting the session must be unlocked manually
-- since it would fail instead overwriting
session.unlock(key())
```

The top() function is a complex thing. I won't describe it in a complete way, but I suggest you to look at the `libero.lua` plugin if the web server that sends you the message source supports the "`Range:`" HTTP request field, or the `tin.lua` plugin if the server needs to be interrupted in a bad way. Remember that the `top()` needs someone that counts the lines and here we have again the stringhack module that counts and may purge some lines.

The javascript is the hell of webmails. Javascripts can do anything and you have to read them to emulate what they do. For example they may add some cookies (and you'll have to do this by hand with the `b:add_cookie()` as in tin.lua) or they may change some form fields (like in the `libero.lua` login load balancing code).

The cookies are sweet enough for us, since the browser module will handle them for us.

The standard files are really system dependent. Under Windows you'll have to constantly look at the `stderr.txt` and `stdout.txt`, while under Unix you will just have to start it with the `-w` switch and look at the console.

The brute force is called Ethereal. Sometimes things don't work in the right way and the only way to debug them is to activate curl debugging to see what FreePOPs does (`b.curl:setopt(curl.OPT_VERBOSE,1)`) and sniff what a real browser does with a tool like Ethereal.

The open source way is the best way of having a good quality piece of software. This means you'll have to release really often your plugin in the development phase and interact much with your testers. Trust me it works, or read "*The cathedral and the bazaar*" by Eric Raymond.

The mimer module is really beta at the time of this tutorial, but is what you need if you are in the unlucky case of a webmail that has no save message button. The `lycos.lua` plugin is an example of what it can do. The main interesting function is `mimer.pipe_msg()` that takes a message header, a text body (in html o plain text format) and some attachments URIs, that are downloaded on the fly, composed into a proper mail message and piped to the mail client.

Parameters to modules may be passed from the `config.lua` file or on the fly using `user@domain?param1=value1&...&paramN=valueN` as described in the plugins chapter. For the plugin writer there is no difference between the two passing mechanisms. The parameters are available to the plugin in the table `freepops.MODULE_ARGS`.

Regex to define handled domains are allowed since version 0.0.29. Official plugins can have a config.lua line like

```
freepops.MODULES_MAP["foo2.*"]  = {
name="foo.lua",
regex = true -- enables the regex processing
}
```

while unofficial plugins can declare a regexp list in the `PLUGIN_REGEXES` field. For example

```
PLUGIN_REGEXES = {"@foo2.*", "@foo3.*", "@foo4[A-Z]*"}
```

8 Submitting a bug

When you have problems or you think you have found a bug, please follow strictly this *iter*:

1. Update to the most recent version of FreePOPs.
2. Try to reproduce the bug, if the bug is not easily reproducible we are out of luck. Something can still be tried: if the software crashed you could compile it from the sources, install valgrind, run freepopsd with valgrind and hope the error messages are interesting.
3. Clean the log files
4. Start FreePOPs with the -w switch
5. Reproduce the bug
6. Send to the developers the log, plus any other info like your system type and how to reproduce this bug.

9 FAQ

How do I configure FreePOPs?

In normal use conditions you don't need to configure FreePOPs, you just have to change your mail client settings as described in the tutorial. For other use cases we provided specific tutorials. Remember to set the POP3 server address to *localhost* (if you've installed FreePOPs on the same computer where you read your mail, otherwise use the IP address of the computer where FreePOPs is installed), the server port to *2000* (or whatever you chose if you ran FreePOPs with the `-p` option) and to set the username to your full email address (in the form `username@webmail.domain`).

If you continue having problems after reading the manual and the tutorial you can post on the forum at HTTP://freepops.diludovico.it (usually in Italian, but feel free to post in English).

How much does it cost?

FreePOPs is Free Software, Free as in Freedom *and* "Free as in Free Beer". You can download it, use it, and modify it freely but if you care about the four friends who made it you may send them a beer asking for their address via email, or a small donation (through the project website on SourceForge).

I've installed FreePOPs and properly configured my mail client, but I'm unable to send mail...?

FreePOPs helps you only in receiving messages. To send mail you have to use the SMTP server of your network provider. If you don't know what SMTP to use, see your ISP website or call their tech support.

This software is in beta stage.. will I lose my mail?

Nobody guarantees the software they write, even if you pay it a lot. It looks safe enough to us, but we can't guarantee anything (no software is really safe and secure). Besides no one guarantees your mail client works well, so if you're using that...

How can I help the project?

Use the source Luke... Sources are freely available, feel free to send us patches and bug reports.

If you want, you can donate to the project via Sourceforge and Paypal.

When you want to report a bug, check that you are using the latest version (we advise you uninstall an old version before installing a new one) and don't forget to include:

- The version number of your copy of FreePOPs
- The operating system you are running
- Your mail client name and its version number
- Most importantly the log of FreePOPs, generated with the **-w** option (read further if you don't know how to add command line parameters). Make sure that the log doesn't contain any sensitive info you wouldn't like to disclose (to the authors). But don't worry, your password is never recorded in the log, it's substituted by a sequence of nine (9) asterisks, no matter what is its real length.

Where is the log?

It depends on the system you are using. On a GNU/Linux system it is probably in */var/log/freepops* or wherever you specified with the `-l` option. On a Windows system the file log.txt is in the same folder of the FreePOPs executable or wherever you specified with the `-l` option.

Before sending the log make sure that it was generated with the **-vv** or **-w** option (see below). These parameters generate a "verbose log" that makes it much easier for us to discover problems. We suggest you erase the log file, start FreePOPs with the switch and recreate the problem. Also make sure that it doesn't contain sensitive info you wouldn't like to disclose.

How do I change FreePOPs's command line switches?

On a Unix system you should know how to do this, just add the switches to the command you use to run FreePOPs (maybe from a script).

On a Windows system you have two options: you may open the properties of the FreePOPs link in the Start -> Programs -> FreePOPs menu (right click on the link then select "Properties...") and add the switches in the command line in the "Destination" box there (after X:\Something\freepopsd.exe); or, run FreePOPs manually from a DOS window with the options you like (again, after the name of the executable).

The command line parameters you need depend on your specific needs. Read the appropriate section of the manual to know what all available parameters are, or run `freepopsd -h` (also `man freepopsd` on Unix).

- Unix example: `/usr/bin/freepopsd -P proxy:port -A user:pass -w -l logfile.txt`

- Windows example: `C:\Programs\FreePOPs\freepopsd.exe -w`

My "AntiVirus" says FreePOPs is a virus!

Stop using that crazy antivirus :-) Seriously, FreePOPs does not contain viruses, trojan horses, worms, arcane formulae for daemonic rites, secret plans for taking over the world or anything like all that. If you don't believe us, the source code is available to everyone (the program is released under the GNU GPL license) and thinking you could hide malicious code in plain sunlight would be crazy. Therefore any antivirus software that detects this kind of problems has accuracy problems of its own, to say the least.

It sure is true that downloading a pre-compiled binary from an unknown source is like asking for trouble. Pre-compiled binary packages you find on the official website (http://www.freepops.org) come from the source code that everyone can get. What you find on free (as in "free beer") software websites or you download through peer-to-peer file sharing systems can't obviously guarantee any kind of safety. So use some common sense.

10 Authors

This manual has been written by Enrico Tassi <gareuselesinge [at] users.sourcefor
and revised and translated by Nicola Cocchiaro <ncocchiaro [at] users.sourceforge.

10.1 Developers

FreePOPs is developed by:

- Enrico Tassi <gareuselesinge [at] users.sourceforge.net>
- Alessio Caprari <alessiofender [at] users.sourceforge.net>
- Nicola Cocchiaro <ncocchiaro [at] users.sourceforge.net>
- Simone Vellei <simone_vellei [at] users.sourceforge.net>

yahoo.lua, hotmail.lua, aol.lua, netscape.lua, mailcom.lua, juno.lua, mail2world.lua, criticalpath.lua, fastmail.lua are developed by:

- Russell Schwager <russells [at] despammed.com>

gmail.lua is developed by:

- Rami Kattan <rkattan [at] gmail.com>
  HTTP://www.kattanweb.com/rami

squirrelmail.lua, tre.lua are developed by:

- Eddi De Pieri <dpeddi [at] users.sourceforge.net>

supereva.lua is developed by:

- Andrea Dalle Molle <Tund3r [at] fastwebnet.it>

LiberoPOPs was developed by:

- Enrico Tassi <gareuselesinge [at] users.sourceforge.net>
- Alessio Caprari <alessiofender [at] users.sourceforge.net>
- Nicola Cocchiaro <ncocchiaro [at] users.sourceforge.net>
- Simone Vellei <simone_vellei [at] users.sourceforge.net>
- Giacomo Tenaglia <sonicsmith [at] users.sourceforge.net>

11 Thanks

Special thanks go to the users who tested the software, to the hackers who made it possible to have a free and reliable development environment as the Debian GNU/Linux system.

Appendix

Since the ASCII table is really common, we have not included one here. You may ask google.