

GNUYahoo! version 1.0.1

User/Developers guide, 10 November 2002

K. Viswanathan gnuvisu@yahoo.com
M.P. Anand Babu ab@gnu.org.in

Copyright © 2002 Free Software Foundation, Inc.

This is the first edition of the GNUYahoo! documentation.

Published by the Free Software Foundation
59 Temple Place - Suite 330,
Boston, MA 02111-1307 USA

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions, except that this permission notice may be stated in a translation approved by the Free Software Foundation.

Table of Contents

1	Overview	1
2	Invoking	2
3	Gnuyahoo commands	3
3.1	Gnuyahoo command - *	3
3.2	Gnuyahoo command - <buddy>	3
3.3	Gnuyahoo command - add	3
3.4	Gnuyahoo command - alias	4
3.5	Gnuyahoo command - bell	4
3.6	Gnuyahoo command - broadcast	4
3.7	Gnuyahoo command - buzz	4
3.8	Gnuyahoo command - forward	5
3.9	Gnuyahoo command - cc	5
3.10	Gnuyahoo command - color-*	5
3.11	Gnuyahoo virtual conference commands	6
3.12	Gnuyahoo command - date	6
3.13	Gnuyahoo command - eval	7
3.14	Gnuyahoo command - gnuyahoo	7
3.15	Gnuyahoo command - help	7
3.16	Gnuyahoo command - ignore*	7
3.17	Gnuyahoo command - load	8
3.18	Gnuyahoo command - ping	8
3.19	Gnuyahoo command - quit	8
3.20	Gnuyahoo command - refresh	8
3.21	Gnuyahoo command - reject	8
3.22	Gnuyahoo command - remove	8
3.23	Gnuyahoo command - restart	9
3.24	Gnuyahoo command - send	9
3.25	Gnuyahoo command - send-file	9
3.26	Gnuyahoo command - sh	9
3.27	Gnuyahoo command - status	10
3.28	Gnuyahoo command - toggle	10
	3.28.0.1 AUTO-INSERT mode HOW-TO	11
3.29	Gnuyahoo virtual conference commands	11
3.30	Gnuyahoo command - who	12
3.31	Gnuyahoo command - xmessage	12
4	Customizing Gnuyahoo	13
4.1	gnuyahoo.scm	13
4.2	init.scm	14
4.3	Default Scheme extensions	14

5	Tips and Tricks	15
	5.0.1 Cursor motion	15
	5.0.2 Editing	15
	5.0.3 Case change	15
6	Extension language	16
7	hello.scm extension	17
	7.1 Writing hello.scm	17
	7.2 Loading hello.scm	17
8	Variables	18
9	Procedures	19
	9.1 General procedures	19
	9.2 Configuration procedures	20
	9.3 Hook related procedures	22
	9.4 Utility procedures	23
10	Hooks	25
11	Learning further	27
12	Authors	28
13	URLs	29
14	Guidelines for submitting a patch	30
15	Portability	31
16	License	32
	Concept Index	33
	Command Index	34
	Procedure Index	36

1 Overview

Gnuyahoo is a freely available GNU messenger for Yahoo! services. It has many features, but to highlight,

- Highly extensible through ‘**Scheme**’ language. (see Chapter 4 [Customization], page 13) (see Chapter 6 [Extension language], page 16)
- Console based client with Readline interface featuring command line editing, history, etc . . . (see Chapter 5 [Tips and Tricks], page 15).
- Most of the features in Gnuyahoo are fully customizable, either through command line arguments (see Chapter 2 [Invoking], page 2), or startup file (see Section 4.1 [gnuyahoo.scm], page 13)
- Consists of almost all the features as the conventional Yahoo messenger for example email alert, conference, ignore etc . . .
- Additional features like alias, bell, forward, cc, eval, gnuyahoo, gy-conf, load, ping, date, sh etc., (see Chapter 3 [Commands], page 3)
- With a new concept called ‘**dynamic-commands**’, a command can appear and disappear dynamically, based on the context.
- AUTO-INSERT feature magically decides and inserts the target buddy name each time you press *RET* during a session. (see Section 3.28 [AUTO-INSERT mode HOW-TO], page 11)
- **history** feature records all your conversions in ‘~/**.gnuyahoo/history/login-id/buddy-name**’.
- Finally, Gnuyahoo is free software. This means that everyone may use it, redistribute it and/or modify it under the terms of the GNU General Public License, as published by the Free Software Foundation (see Chapter 16 [License], page 32)

2 Invoking

Invoking Gnuyahoo at command prompt is very simple. The following are the possible command-line arguments supported,

gnuyahoo [*options*]
where *options* are,

- `-u=yahoo-id` | `—user=yahoo-id`
yahoo-id is your yahoo account name.
- `-s=status` | `—status=status`
status can be one of the following numbers,
 - 0: I'm Available
 - 1: Be Right Back
 - 2: Busy
 - 3: Not at Home
 - 4: Not at my Desk
 - 5: Not in the Office
 - 6: On the Phone
 - 7: On Vacation
 - 8: Out to Lunch
 - 9: Stepped Out
 - 12: Invisible
 - 999: Idle
- `-v` | `—version`
Gives the current version of Gnuyahoo. This option does not accept any argument.
- `-h` | `—help`
Gives a brief help on the above options. This option does not accept any argument.

3 Gnuyahoo commands

3.1 Gnuyahoo command - *

***** *message* command

Broadcast *message* to all buddies in the list. * is a virtual buddy in your buddy list which means everybody in the list.

```
GY!> * Hello, World!
```

```
GY!> *
```

The message `Hello, World!` will be sent to all the buddies in your buddy list.

3.2 Gnuyahoo command - <buddy>

<buddy> *message* command

Send *message* to *buddy*. You can use *TAB* key to fill the *buddy* name automatically.

On conflict press *TAB* twice to list the conflicting *buddy* names.

Example: A sample Gnuyahoo session.

```
GY!> ?who
```

```
[Friends]
```

```
* markus [Busy Hacking]
```

```
  rms
```

```
* thomas [Idle]
```

```
[Team]
```

```
  balugi
```

```
* kvisu2000
```

```
GY!> kvisu2000 Hi, How are you
```

```
kvisu2000 -> I'm fine
```

```
GY!> balugi Hi, Are you there
```

```
Offline message sent to [balugi]
```

3.3 Gnuyahoo command - add

?add *buddy* [*group*] [*message*] command

Add a buddy to your buddy list. The following are the possible arguments for this command,

- a. *buddy* must be a Yahoo ID.
- b. Optional *group* under which the *buddy* is added. If the *group* does not exist it will be created newly. The default *group* is `Friends`.
- c. *message* is optional message.

3.4 Gnuyahoo command - alias

?alias *name buddy1 [buddy2 buddy3 ...]* command

This command expands *name* to *buddy1*, *buddy2* etc. **?alias** accepts *name* and at least one *buddy* as arguments. Aliases can be recursive.

Using Guile interface, you can add permanent aliases to startup file (see Section 4.1 [gnuyahoo.scm], page 13) like,

```
(define alias '((helpdesk . (abindian balugi kvisu2000))
               (mridul . (gnuindian))
               (bala . (balugi))
               (nags . (nagappanal))
               (visu . (kvisu2000))))
```

3.5 Gnuyahoo command - bell

?bell command

This command switches bell sound between ON and OFF. By default 'bell' is ON. A better interface to this command is **?toggle** (see [?toggle bell], page 10)

Using Guile interface, you can disable bell during startup (see Section 4.1 [gnuyahoo.scm], page 13) like,

```
(gy-bell!)
```

3.6 Gnuyahoo command - broadcast

broadcast *message* command

Broadcast *message* to all buddies in the list. This is same as * (see Section 3.1 [<star>], page 3).

```
GY!> ?broadcast Hello, World!
GY!>
```

The message `Hello, World!` will be sent to all the buddies in your buddy list.

3.7 Gnuyahoo command - buzz

buzz *buddy* command

send a BUZZ! to *buddy*

```
GY!> ?buzz marcus
GY!>
```


3.8 Gnuyahoo command - forward

?forward *from-buddy to-buddy1 [to-buddy2 to-buddy3 ...]* command

Messages received from *from-buddy* are forwarded to *to-buddy1*, *to-buddy2* etc.

?forward accepts *name* and at least one *to-buddy* as arguments.

Using Guile interface, you can add permanent forwards to startup file (see Section 4.1 [gnuyahoo.scm], page 13) like,

```
(define forward '((gnubot . (ramyog_2000 nagappanal balugi))
                 (gopal_narayanan . (parag_mehta))))
```

3.9 Gnuyahoo command - cc

?cc *buddy cc-buddy1 [cc-buddy2 cc-buddy3 ...]* command

Message sent to *buddy* is CC'ed to *cc-buddy1*, *cc-buddy2* etc. ?cc accepts *buddy* and at least one *cc-buddy* as arguments.

Using Guile interface, you can add permanent CCs to startup file (see Section 4.1 [gnuyahoo.scm], page 13) like,

```
(define cc '((rms . (markus roland thomas))
            (gopal_narayanan . (parag_mehta))))
```

3.10 Gnuyahoo command - color-*

?color-on command

Enables color themes mode.

Using Guile interface, you can permanently enable color themes in startup file (see Section 4.1 [gnuyahoo.scm], page 13) like,¹

```
(gy-enable-colors)
```

?color-off command

Disables color themes mode.

Using Guile interface, you can permanently disable color themes in startup file (see Section 4.1 [gnuyahoo.scm], page 13) like,

```
(gy-disable-colors)
```

¹ by default color themes is enabled

?color-buddy *buddy color* command

Displays all messages from *buddy* in the specified *color*. Possible *color* values are [red, blue, yellow, magenta, green, cyan, white].

Using Guile interface, you can permanently set color for a buddy message in startup file (see Section 4.1 [gnuyahoo.scm], page 13) like,

```
(gy-set-buddy-color! "nirranjan" "green")
(gy-set-buddy-color! "rms" "red")
(gy-set-buddy-color! "marcus" "magenta")
```

3.11 Gnuyahoo virtual conference commands

This is standard yahoo conference this feature is virtual.

?conf-start *room buddy1 [buddy2 buddy3 ...]* command

This command starts a conference with room name *room* with buddies *buddy1*, *buddy2* etc.

?conf-add *room buddy* command

This command add a buddy, to a existing conference room.

?conf-list command

This command lists all the conference rooms associated with you currently.

?conf-end *room* command

This command quits you from a conference rooms.

?conf-decline *room* command

This command send your decline to a received invitation from a conference rooms.

?conf-send *room* command

This command sends message to a conference room.

3.12 Gnuyahoo command - date

?date [*arguments*] command

This command displays the system date. Try ‘--help’ for complete list of *arguments*.

3.13 Gnuyahoo command - eval

?eval *exp* command

Evaluate *exp*, a list representing a Scheme expression. You have complete access to gnuyahoo-guile internals, including Scheme extensions through this command.

Example: To send message to yourself

```
GY!> ?eval (gy-send-message (gy-get-default-login-id) "Hello GNU")
abindian -> Hello GNU
GY!>
```

3.14 Gnuyahoo command - gnuyahoo

?gnuyahoo *buddy* command

This command checks if *buddy* is using Gnuyahoo.

Example: To check if 'kvisu2000' is using Gnuyahoo,

```
GY!> ?gnuyahoo kvisu2000
Yes [kvisu2000] is using GNUYahoo
GY!>
```

3.15 Gnuyahoo command - help

?help [*command*] command

You can always ask Gnuyahoo itself for information on its commands, using the command **?help**. If *command* is ignored then help on all the commands will be listed.

3.16 Gnuyahoo command - ignore*

?ignore *buddy* command

Adds this BUDDY to Yahoo ignore list. This *buddy* will be removed from the buddy list and you will get no messages from him/her.

?unignore *buddy* command

This command unignores *buddy* from the Yahoo ignore list.

?ignore-list command

Displays the Yahoo *ignore-list*

3.17 Gnuyahoo command - load

?load *scheme-file* command

?load command loads and evaluates Scheme extensions. *scheme-file* argument is a must.

Using Guile interface, you can load other Scheme files from startup file (see Section 4.1 [gnuyahoo.scm], page 13) like,

```
(load "/home/gnu/hello.scm")
```

(see [gy-load], page 19)

3.18 Gnuyahoo command - ping

?ping *buddy count* command

Send ping messages to a *buddy*, *count* times.

3.19 Gnuyahoo command - quit

?quit command

Logout and exit from Gnuyahoo.

3.20 Gnuyahoo command - refresh

?refresh command

This command refreshes the buddy list and the status information.

3.21 Gnuyahoo command - reject

?reject command

This command rejects the buddy for adding you in his/her buddy list and removes your name from his/her buddylist.

3.22 Gnuyahoo command - remove

?remove *buddy* [*message*] command

This command permanently removes the BUDDY from buddy list. *message* argument is optional.

Example:

```
GY!> ?remove balugi Poda Kuppa!
```

3.23 Gnuyahoo command - restart

?restart command
 This command restarts Gnuyahoo from inside Gnuyahoo.

3.24 Gnuyahoo command - send

?send *buddy message* command
 This command sends *message* to the *buddy*.
 Example:

```
GY!> ?send balugi Hi, how are you?
balugi -> Fine and you?
```

3.25 Gnuyahoo command - send-file

?send-file *buddy filepath [message]* command
 This command sends *file* to the *buddy*.
 Example:

```
GY!> ?send-file balugi /etc/passwd my passwd file
```

3.26 Gnuyahoo command - sh

?sh [*command*] [*args ...*] command
 ?sh command executes the specified *command* with its *args*. With no arguments ?sh escapes to shell.

Example:

```
GY!> ?sh ls -lh /tmp
total 12k
drwxr-xr-x    3 root   root    4.0k Jan  1 00:53 emacs-terminfo
-rw-r--r--    1 root   root      1 Jan  1 05:04 emacsOdVut8
drwx-----  2 root   root    4.0k Jan  1 00:13 xdvi7GIKqr
GY!> ?sh
press C-d to return to Gnuyahoo
$ rm -f /tmp/xdvi7GIKqr
C-d RET
GY!>
```

3.27 Gnuyahoo command - status

?status [*status-number*] [*custom-message*] command

Using **?status** command, you can view or set your buddy-status.

- a. *status-number* should denote one of the following.

```

0: I'm Available
1: Be Right Back
2: Busy
3: Not at Home
4: Not at my Desk
5: Not in the Office
6: On the Phone
7: On Vacation
8: Out to Lunch
9: Stepped Out
12: Invisible
99: [Custom message]
999: Idle

```

- b. Optionally you can mention *custom-message*, if *status-number* is 99.
c. When no arguments are supplied, **?status** command displays your current status.

3.28 Gnuyahoo command - toggle

?toggle *state* command

This command toggles the following *states*.

- **bell**
Toggle *bell* sound between ON and OFF. Default is ON.
Example:

```

GY!> ?toggle bell
Bell sound - [OFF]
GY!> ?toggle bell
Bell sound - [ON]
GY!>

```
- **session**
Toggle *session* mode between VANILLA and AUTO-INSERT. Default is AUTO-INSERT.
VANILLA mode lets the user to type the buddy name manually. However user can use *TAB* interface to auto-fill. (see Chapter 5 [Tips and Tricks], page 15)
AUTO-INSERT mode intelligently selects the buddy name during chat session. (see Section 3.28 [AUTO-INSERT mode HOW-TO], page 11).

2

² Currently typing notifications will be sent remote buddy only in AUTO-INSERT mode. VANILLA mode has no concept of current target buddy.

- **status**
Toggle display of *status* change notifications between SHOW and HIDE. Default SHOW.
- **who**
Toggle *who* mode between ONLINE-ONLY and SHOW-ALL. Default is ONLINE-ONLY. (see Section 3.30 [who], page 12).

3.28.0.1 AUTO-INSERT mode HOW-TO

AUTO-INSERT mode makes Gnuyahoo intelligent by automatically selecting the buddy name, every time when the user types the message. The following is a small HOW-TO on AUTO-INSERT mode,

- How AUTO-INSERT guesses the buddy name?
AUTO-INSERT mode guesses the buddy name from the previously sent/received message.
- How to change the target buddy selected by AUTO-INSERT mode?
Just move the cursor back and edit the buddy name to your choice.
- How to switch the buddy name to reply to the last received message?
Pressing *RET* without typing any message switches to last received buddy.

3.29 Gnuyahoo virtual conference commands

Unlike standard yahoo conference this feature is virtual. Its actually a combination of *cc* and *forward* extensions. All the messages you send to a virtual buddy named *cf* are despatched to the conference members. Similarly the messages received from any of the conference members are forwarded back to other conference members.

?vconf-start *buddy1* [*buddy2 buddy3 . . .*] command

This command starts a virtual conference with *buddy1*, *buddy2* etc. On conference start **?vconf-start** disappears and **?vconf-who**, **?vconf-end** appears. **?vconf-start** accepts at least one *buddy* as its argument.

?vconf-who command

This command lists all the conference members.

?vconf-end command

This command ends the virtual conference. On conference end **?vconf-who**, **?vconf-end** disappears and **?vconf-start** appears.

3.30 Gnuyahoo command - who

?who command

This command displays the buddy list as well as their current buddy status.

Example:

```
GY!> ?who

[Friends]
* markus [Busy Hacking]
  rms
* thomas [Idle]

[Team]
  balugi
* kvisu2000

GY!>
```

3.31 Gnuyahoo command - xmessage

?xmessage *buddy message* command

Send a X popup *message* to *buddy*.

Example:

```
GY!> ?xmessage bala Hi Bala, are you there
```


4 Customizing Gnuyahoo

Gnuyahoo can be customized to a great extent using Guile interface. User can himself customize or extend new features in Gnuyahoo using Scheme as extension language. Most of the features are already written in Scheme.

If you want to extend Gnuyahoo yourself, you can further explore Gnuyahoo Extension Developer Guide. (see Chapter 6 [Extension language], page 16)

4.1 gnuyahoo.scm

Gnuyahoo loads the startup options from ‘~/gnuyahoo/gnuyahoo.scm’. Right from custom settings like username, password ... to complete Scheme programming can be done in this file.

```

; this is comment
;;; sample gnuyahoo.scm
;;; login-id is equal to my gnu/linux account
;; (gy-set-default-login-id! (array-ref (getpwuid (getuid)) 0))

;;; default login-id for yahoo service
(gy-set-default-login-id! "gnu_india")
;; (gy-set-default-login-id! "abindian")
;; (gy-set-default-login-id! "gnubot")

;;; default global password
(gy-set-default-password! "nopassword")
;; (gy-set-default-password! "presenter")

;;; by default session mode is AUTO-INSERT. switch it to VANILLA mode
;; (gy-toggle! "session")

;;; by default display of status message is SHOW. switch it HIDE
;; (gy-toggle! "status")

;;; if login-id is "abindian"
(and (string=? (gy-get-default-login-id) "abindian")
     ;;; default password
     (gy-set-default-password! "presenter")
     ;;; switch OFF bell
     (gy-toggle! "bell")
     ;;; show ALL buddies
     (gy-toggle! "who")
     ;;; login in invisible mode
     (gy-set-default-status! 12))

;;; if login-id is "gnubot"
(and (string=? (gy-get-default-login-id) "gnubot")
     ;;; default password
     (gy-set-default-password! "pressescape"))

```

```

;;; switch OFF bell
(gy-toggle! "bell")

;;; create aliases
(define alias '((helpdesk . (abindian balugi kvisu2000))
              (mridul . (gnuindian))
              (bala . (balugi))
              (nags . (nagappanal))
              (visu . (kvisu2000))))

;;; create CC lists
(define cc '((rms . (markus roland thomas))
           (gopal_narayanan . (parag_mehta))))

;;; create forward lists
(define forward '((gnubot . (ramyog_2000 nagappanal balugi))
                (gopal_narayanan . (parag_mehta))))

```

All entries in this ‘gnuyahoo.scm’ file are optional. However there is no limit in customizing or extending Gnuyahoo through Guile interface. Explaining all the possibilities are beyond the scope of this document.

(see Chapter 6 [Extension language], page 16)

4.2 init.scm

All system wide policy settings and extensions are loaded through ‘i.scm’. By default you can find ‘init.scm’ at ‘/usr/share/gnuyahoo/extensions/'. To override this system wide ‘init.scm’ file, copy it to ‘~/gnuyahoo/extensions/init.scm’. You must be aware of what you are doing, before you mess up anything here.

(see Chapter 6 [Extension language], page 16)

4.3 Default Scheme extensions

Most of the Gnuyahoo features are available through Scheme extensions. To override these extensions, copy them from ‘/usr/share/gnuyahoo/extensions/' to ‘~/gnuyahoo/extensions/' and edit them.

(see Chapter 6 [Extension language], page 16)

5 Tips and Tricks

You are free to use complete Readline keys inside Gnuyahoo. Frequently used Readline keys inside Gnuyahoo are,

5.0.1 Cursor motion

character	C-b	C-f
word	M-b	M-f
line up/down	C-p	C-n
line start/end	C-a	C-e

5.0.2 Editing

delete char	C-d
delete char backwards	C-h
delete word	M-d
delete word backwards	C-w
kill line	C-k
kill line backwards	C-u
character swap	C-t
word swap	M-w
paste	C-y
undo	C-_
repeat prefix	M-number

5.0.3 Case change

uppercase word	M-u
lowercase word	M-l
capitalize word	M-c

If you want to do further stunts, jump to Readline manual, See section “Readline” in *Readline*.

1

¹ When you press *TAB* twice at Gnuyahoo prompt you can see all the possible commands and buddy names.

6 Extension language

An *extension language* is a programming language interpreter offered by an application program, so that users can write macros or even full-fledged programs to extend the original application. Extension languages have a C interface (it is usually C, but it could be any other compiled language), and can be given access to the C data structures. Likewise, there are C routines to access the extension language data structures.

Gnuyahoo uses GNU extension language - *Guile* (which can stand for `_GNU Ubiquitous Intelligent Language Extension_`). Guile started out as an embeddable Scheme interpreter, and has rapidly evolved into a kitchen-sink package including a standalone Scheme interpreter, an embeddable Scheme interpreter, several graphics options, other languages that can be used along with Scheme (for now just `_ctax_` and `_Tcl_`), and hooks for much more.

7 hello.scm extension

Learn how to write a simple extension by yourself.

7.1 Writing hello.scm

This example extension creates a dynamic command `?hello`, which on invocation sends a message *Hello GNU* to yourself.

```
;; hello.scm
(gy-register-command! "?hello" "?hello\n\t- Hello to myself.\n")
(define (?hello args)
  "send me hello message"
  (gy-send-message (gy-get-default-login-id) "Hello GNU"))
```

7.2 Loading hello.scm

Copy 'hello.scm' to '~/.gnuyahoo/extensions/' and add this entry in your '~/.gnuyahoo/gnuyahoo.s' (gy-load "hello.scm")

8 Variables

Currently no variables are exported to Scheme environment from Gnuyahoo. Instead we have solved such needs using procedure interface to get/set variables.

9 Procedures

The following are the list of gnuyahoo procedures that are exported to Scheme. Now you are able to call the procedures from Scheme that are written in C.

9.1 General procedures

- gy-load** *filepath* primitive
 Loads and evaluates *filepath.scm* from mentioned path or from ‘~/gnuyahoo/extensions/’ or from ‘/usr/share/gnuyahoo/extensions/’.
 Example:

```
(gy-load "aliases.scm")
```
- gy-add-buddy** *buddy group* primitive
 Adds *buddy* to *group* in your contact list.
 Example:

```
(gy-add-buddy "rms" "GNU")
```
- gy-send-message** *buddy message* primitive
 Sends *message* to the *buddy*.
 Example:

```
(gy-send-message "rms" "hello GNU")
```
- gy-send-message-no-hook** *buddy message* primitive
 Sends *message* to the *buddy*. This procedure does not run any hooks. You will have to use this procedure while sending messages from inside `gy-message-send-hook` otherwise it will lead to an endless recursion.
 Example:

```
(gy-send-message-no-hook "rms" "hello GNU")
```
- gy-set-current-target-buddy!** *buddy* primitive
 Sets the target *buddy* name in AUTO-INSERT mode. This call makes meaning only when you are in AUTO-INSERT mode.
 Example:

```
(gy-set-current-target-buddy! "richard")
```
- gy-register-command!** *command documentation* primitive
 Registers a dynamic *command* and its *documentation* with Gnuyahoo’s command interpreter.
 Example:

```
(gy-register-command! "?date" "?date\n\t- display current date")
```

gy-unregister-command! *command* primitive

Un-registers *command* from Gnuyahoo's command interpreter.

Example:

```
(gy-unregister-command! "?date")
```

gy-version primitive

Return the current version string of Gnuyahoo.

Example:

```
(display (gy-version))
```

gy-display *message* primitive

Prints the *message* in the console. Unlike the `display` primitive, this procedure takes care of printing *messages* asynchronously.

Example:

```
(gy-display ("I am proud of Gnuyahoo"))
```

9.2 Configuration procedures

gy-bell primitive

Switches the bell sound between ON and OFF.

Example:

```
(gy-bell)
```

gy-toggle! *state* primitive

Switches *state* between two different states.

The value of *state* can be,

state `bell` Switches bell sound between ON and OFF.

Example:

```
(gy-toggle! bell)
```

state `session`

Switches Gnuyahoo session between AUTO-INSERT and VANILLA modes.

Example:

```
(gy-toggle! session)
```

state `status`

Switches display of status messages between SHOW and HIDE.

Example:

```
(gy-toggle! status)
```


state who Switches who mode between ONLINE-ONLY and SHOW-ALL.

Example:

```
(gy-toggle! who)
```

gy-get-home-dir

primitive

Return the home directory of current user.

Example:

```
(chdir (gy-get-home-dir))
```

gy-get-config-dir

primitive

Return the directory containing configuration files.

Example:

```
(display (gy-get-config-dir))
```

gy-get-config-filename

primitive

Return the configuration filepath.

Example:

```
(display (gy-get-config-filename))
```

gy-get-download-filename

primitive

Return the filename containing URL downloads.

Example:

```
(display (gy-get-download-filename))
```

gy-get-global-extensions-directory

primitive

Return the directory containing global extensions.

Example:

```
(chdir (gy-get-global-extensions-directory))
```

gy-get-local-extensions-directory

primitive

Return the directory containing local extensions.

Example:

```
(chdir (gy-get-local-extensions-directory))
```

gy-get-default-login-id

primitive

Return the yahoo-id of currently logged-in user.

Example:

```
(display (gy-get-default-login-id))
```

gy-set-default-login-id! *yahoo-id* primitive

Sets the *yahoo-id*.

Example:

```
(gy-set-default-login-id! "kvisu2000")
```

gy-set-default-password! *password* primitive

Sets the yahoo account *password*.

Example:

```
(gy-set-default-password! "presenter")
```

gy-set-default-status! *number* primitive

Sets the yahoo status *number*.

Example:

```
;;; go invisible
(gy-set-default-status! 12)
```

For complete list of status number definitions refer (see Section 3.27 [status], page 10)

gy-set-prompt! *prompt-string* primitive

Sets the yahoo prompt.

Example:

```
;;; set gnuyahoo prompt
(gy-set-prompt! "~qp~> ")
```

9.3 Hook related procedures

1

gy-hook-return primitive

Makes the calling procedure return immediately after running the hooks.

Example:

```
(define (alias to message)
  "alias nags to nagappanal"
  (and (string=? to "nags")
       ;; send message to actual name
       (gy-send-message-no-hook "nagappanal" message)
       ;; "nags" doesn't exist. so let send return immediately
       (gy-hook-return)))
(add-hook! gy-message-send-hook alias)
```

¹ The following primitives can be called from procedures that are hooked to Gnuyahoo exported hooks. Also check the list of supported primitives for each hook.

9.4 Utility procedures

These are general purpose utility procedures written completely in Scheme.

²

symbolrnumber->symbol *num* procedure

Converts *num* to its corresponding Scheme symbol. *num* is any number atom in Scheme.

Example:

```
(symbolrnumber->symbol 5)
```

will return 5 which is a Scheme symbol and not a number.

list->asv *list delimiter* procedure

Converts *list* to vector delimited by *delimiter*. *list* stands for list to any seperated vector.

Example:

```
(list->asv ("gnuindian" "nagappanal" "abindian" "balugi") ", ")
```

will return "gnuindian, nagappanal, abindian, balugi"

list->csv *list* procedure

Converts *list* to a comma seperated vector delimited by ', '.

Example:

```
(list->csv ("gnuindian" "nagappanal" "abindian" "balugi"))
```

will return "gnuindian, nagappanal, abindian, balugi"

list->symlist *list* procedure

Converts *list* of numbers/symbols into a list of symbols.

Example:

```
(list->symlist (5 a gnu 100 10.5))
```

will return (5 a gnu 100 10.5) where all items in the list are symbols and not numbers.

List already containing symbols are not altered.

send-message-to-group *group message* procedure

Send *message* to a *group*.

Example:

```
(send-message-to-group ("gnuindian" "nagappanal" "abindian" "balugi") "Hello G
```

² utility procedures are loaded through /DATADIR/gnuyahoo/extensions/util.scm

local-date-time

Returns the local date and time

Example:

```
(display (local-date-time))
```

procedure

ignore-message! *message-pattern*

ignores messages matching the regex *message-pattern*

Example:

```
(ignore-message! "^PING$")
```

procedure

ignored-message? *message*

Return #t if this *message* is ignored, else #f.

Example:

```
(ignored-message? "PING")
```

procedure

10 Hooks

Through Hooks facility Gnuyahoo lets you steal its control at various important junctures during execution.

gy-message-send-hook *buddy message* hook

Hook procedure is called with *buddy* and *message* as arguments on every send message operation.

Supporting primitives:

see [gy-hook-return], page 22

Example:

```
(define (cc-proc to message)
  "hook procedure for CCing messages"
  (and (string=? to "rms")
       (gy-send-message-no-hook "thomas" message)
       (gy-send-message-no-hook "roland" message)
       (gy-send-message-no-hook "gord" message)))
(add-hook! gy-message-send-hook cc-proc)
```

gy-message-receive-hook *buddy message* hook

Hook procedure is called with *buddy* and *message* as arguments on every receive message operation.

Supporting primitives:

see [gy-hook-return], page 22

Example:

```
(define (forward-proc from message)
  "hook procedure for bouncing messages"
  (and (string=? from "rms")
       (gy-send-message-no-hook "thomas" message)
       (gy-send-message-no-hook "roland" message)
       (gy-send-message-no-hook "gord" message)))
(add-hook! gy-message-receive-hook forward-proc)
```

gy-message-receive-offline-hook *buddy message timestamp* hook

Hook procedure is called with *buddy*, *message* and *timestamp* as arguments on every receive offline message operation.

Supporting primitives:

see [gy-hook-return], page 22

Example:

```
(define (ack-proc from message timestamp)
  "hook procedure for acknowledging offline messages"
  (gy-send-message-no-hook from "Received ur offline message"))
(add-hook! gy-message-receive-offline-hook ack-proc)
```

gy-login-post-hook

hook

Hook procedure is called after completion of login operation.

Supporting primitives:

see [gy-hook-return], page 22

Example:

```
(define (history-rotate-proc)
  "hook procedure for flushing the old history messages"
  (history-rotate))
(add-hook! gy-login-post-hook history-rotate-proc)
```

gy-contact-added-hook

hook

Hook procedure is called after a buddy adds you in his/her contact list.

Supporting primitives:

see [gy-hook-return], page 22

Example:

```
(define (auto-add-buddy from message)
  "hook procedure for automatically adding buddy to your contact list"
  (gy-add-buddy from "GNU")) ;; GNU - group name
(add-hook! gy-contact-added-hook auto-add-buddy)
```

11 Learning further

The following are the URLs where you can find useful manuals for Guile and Scheme.

<http://www.gnu.org/software/guile/>
<http://www.schemers.org/>
ftp://ftp.cs.utexas.edu/pub/garbage/cs345/schintro-v14/schintro_toc.html
<http://www.informatik.uni-kiel.de/~scheme/>
<http://freespace.virgin.net/david.drysdale/guile/tutorial.html>
http://nis-www.lanl.gov/~rosalia/gnudl-doc/learn_libguile_toc.html
http://theoryx5.uwinnipeg.ca/gnu/guile/guile-user.html#SEC_Top
http://www.nada.kth.se/~mdj/guile-ref/guile-ref_toc.html
<http://www.red-bean.com/guile/guile/old/3540.html>
<http://nis-www.lanl.gov/~rosalia/mydocs/guile-user.html>
<http://www.cs.utexas.edu/users/lavender/courses/scheme/>
http://www.cstr.ed.ac.uk/projects/festival/manual/festival_8.html#SEC24
<http://www.cs.ccu.edu.tw/~dan/tutorials.html>
<http://www.wcug.wvu.edu/~randyman/COMPUTERS/SCHEME/start.htm>
<http://www.dmoz.org/Computers/Programming/Languages/Lisp/Scheme/Teaching/>
<http://www.cs.caltech.edu/~cs181/doc/>
<http://cis.csuohio.edu/~hysocel/Links/Documents.html>

12 Authors

We believe in Software Freedom and Ethics, the GNU's way.

- | | | | |
|----|-----------------|-----------------|-------------------------------------|
| 1. | A. Balamurugan | Developer | <code>balamurugana@yahoo.com</code> |
| 2. | A. Nagappan | Developer | <code>nagappanal@yahoo.com</code> |
| 3. | Gopal Narayanan | Developer | <code>gopal@debian.org</code> |
| 4. | H.E. Ramesh | Developer | <code>ramyog_2000@yahoo.com</code> |
| 5. | K. Nirranjan | Developer | <code>nirranjan@yahoo.com</code> |
| 6. | K. Viswanathan | Developer | <code>gnuvisu@yahoo.com</code> |
| 7. | M.P. Anand Babu | Project Manager | <code>ab@gnu.org.in</code> |
| 8. | Mridul Jain | Developer | <code>gnuindian@yahoo.com</code> |
| 9. | Parag Mehta | Developer | <code>pm@gnuos.org</code> |

Texi documentation written by K. Viswanathan `gnuvisu@yahoo.com` and revised by M.P. Anand Babu `ab@gnu.org.in`

13 URLs

Homepage

URL

<http://gnuyahoo.sourceforge.net/>

Download

URL

http://sourceforge.net/project/showfiles.php?group_id=28440&release_id=54334

CVS

URL

To know project information like Bugs, Updates, Support, Patches, Tasks, News, Development Status, Activity Percentile, Project Activity visit <http://sourceforge.net/projects/gnuyahoo/>

Mailing list

URL

Gnuyahoo has its own mailing list. The mailing list is for discussion of features, reporting bugs (those that you think may be of interest to the public) and mailing announcements. You are welcome to subscribe. The more people on the list, the better!

To Subscribe/Unsubscribe visit, <http://lists.sourceforge.net/lists/listinfo/gnuyahoo-bugs>

Post messages to mailing list at gnuyahoo-bugs@lists.sourceforge.net.

For mailing list archives visit, <http://www.geocrawler.com/lists/3/SourceForge/13348/0/>

Bugs

URL

You are welcome to send bug reports about Gnuyahoo to gnuyahoo-bugs@lists.sourceforge.net. The bugs that you think are of the interest to the public (i.e. more people should be informed about them) can be Cc-ed to the above mailing lists.

Before actually submitting a bug report, please try to follow a few simple guidelines.

1. Please try to ascertain that the behavior you see really is a bug. If Gnuyahoo crashes, it's a bug. If Gnuyahoo does not behave as documented, it's a bug. If things work strange, but you are not sure about the way they are supposed to work, it might well be a bug.
2. Try to repeat the bug in as simple circumstances as possible.
3. If Gnuyahoo has crashed, try to run it in a debugger, e.g. 'gdb 'which gnuyahoo' core' and type **where** to get the backtrace.
4. Find where the bug is, fix it and send the patches. :) (see Chapter 14 [Guidelines for submitting a patch], page 30)

Send your specific queries to M.P. Anand Babu ab@gnu.org.in

14 Guidelines for submitting a patch

1. Copy the latest CVS version of `gnuyahoo` directory as `gnuyahoo-hack`.
2. Make changes in your `gnuyahoo-hack` directory.
3. Create patch using

```
# diff -pruN gnuyahoo gnuyahoo-hack > gnuyahoo-patch-title
```
4. Mail the patch file '`gnuyahoo-patch-title`' to the mailing list with subject prefixed with '`PATCH:`'.
Please send only text mails with patch as a part of the message body. Don't update '`ChangeLog`' file, instead add your comments at the beginning of the body.

15 Portability

Since Gnuyahoo uses GNU Autoconf for building and configuring, and avoids using ‘`special`’ ultra-mega-cool features of any particular Unix, it should compile (and work) on all common Unix flavors.

Various Gnuyahoo versions have been compiled and tested under GNU/Hurd and GNU/Linux. However Gnuyahoo can be easily ported to any POSIX complaint platform with Guile and Readline ports. If you compile it on an architecture not listed here, please let us know so that we can update it. (see Chapter 13 [URLs], page 29)

16 License

The license of Gnuyahoo consists of the GNU GPL plus a special statement giving blanket permission to link with non-free software. This is the license statement as found in any individual file that it applies to:

```
This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2, or (at your option)
any later version.
```

```
This program is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this software; see the file COPYING. If not, write to the
Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA
02111-1307 USA
```

```
As a special exception, the Free Software Foundation gives permission
for additional uses of the text contained in its release of Gnuyahoo.
```

```
The exception is that, if you link the Gnuyahoo with other files to
produce an executable, this does not by itself cause the resulting
executable to be covered by the GNU General Public License. Your use
of that executable is in no way restricted on account of linking the
Gnuyahoo code into it.
```

```
This exception does not however invalidate any other reasons why the
executable file might be covered by the GNU General Public License.
```

```
This exception applies only to the code released by the Free Software
Foundation under the name Gnuyahoo. If you copy code from other Free
Software Foundation releases into a copy of Gnuyahoo, as the General
Public License permits, the exception does not apply to the code that
you add in this way. To avoid misleading anyone as to the status of
such modified files, you must delete this exception notice from them.
```

```
If you write modifications of your own for Gnuyahoo, it is your choice
whether to permit this exception to apply to your modifications. If
you do not wish that, delete this exception notice.
```

Concept Index

A

AUTHORS	28
AUTO-INSERT	11

C

Command line arguments	2
Customization	13

E

Extension language	16
--------------------------	----

F

Features	1
----------------	---

G

Gnuyahoo URLs	29
gnuyahoo.scn	14
Guile	16

I

init.scn	14
Initialization	14

O

ONLINE-ONLY	11
Overriding extensions	14

P

Portability	31
-------------------	----

R

Readline in Gnuyahoo	15
----------------------------	----

S

SHOW-ALL	11
Startup	14
Submitting patches	30

V

VANILLA	11
---------------	----

Command Index

*		H	
*	3	help	7
<		I	
<buddy>	3	ignore	7
A		L	
add	3	load	8
alias	4	P	
B		ping	8
bell	4	Q	
broadcast	4	quit	8
buzz	4	R	
C		refresh	8
cc	5	reject	8
color-buddy	6	remove	8
color-off	5	restart	9
color-on	5	S	
conf-add	6	send	9
conf-begin	6	send-file	9
conf-decline	6	sh	9
conf-list	6	status	10
conf-quit	6	T	
conf-send	6	toggle	11
D		U	
date	6	unignore	7
E		V	
eval	7	vconf-end	11
F		vconf-start	11
forward	5	vconf-who	11
G			
gnuyahoo	7		

W

who 12

X

xmessage 12

Procedure Index

G

- gy-add-buddy 19
- gy-bell 20
- gy-contact-added-hook 26
- gy-display 20
- gy-get-config-dir 21
- gy-get-config-filename 21
- gy-get-default-login-id 22
- gy-get-download-filename 21
- gy-get-global-extensions-directory 21
- gy-get-home-dir 21
- gy-get-local-extensions-directory 21
- gy-hook-return 22
- gy-load 19
- gy-login-post-hook 26
- gy-message-receive-hook 25
- gy-message-receive-offline-hook 26
- gy-message-send-hook 25
- gy-register-command! 20
- gy-send-message 19
- gy-send-message-no-hook 19
- gy-set-current-target-buddy! 19
- gy-set-default-login-id! 22

- gy-set-default-password! 22
- gy-set-default-status! 22
- gy-set-prompt! 22
- gy-toggle! 21
- gy-unregister-command! 20
- gy-version 20

I

- ignore-message! 24
- ignored-message? 24

L

- list->asv 23
- list->csv 23
- list->symlist 23
- local-date-time 24

S

- send-message-to-group 23
- symbolrnumber->symbol 23