

# **POLChat Client**

---

Reference Manual for version 2.0

**Marek Futrega**

---



# Table of Contents

<b>1</b>	<b>General Information</b>	<b>1</b>
1.1	Basic Concepts	1
1.2	Main Features	1
<b>2</b>	<b>Installation</b>	<b>2</b>
<b>3</b>	<b>Client Customization</b>	<b>3</b>
3.1	Logging In	3
3.2	Components	4
3.3	URLs	5
3.4	Preferences	5
3.5	Colors	6
3.6	Fonts	6
3.7	Connections	7
<b>4</b>	<b>Using the Chat</b>	<b>8</b>
4.1	Entering the Chat	8
4.2	Joining a Room	8
4.3	Posting Messages	9
4.4	Ignoring Users	10
4.5	Buddies	10
4.6	Quitting the Chat	11
4.7	Miscellaneous Commands	11
4.8	Preferences Window	11
4.9	Rooms Window	12
<b>5</b>	<b>Using the Chat as an Operator</b>	<b>13</b>
5.1	Room Operators	13
5.1.1	Managing Room Operators	13
5.1.2	Kicking Users Out of the Room	14
5.1.3	Banning Users from the Room	15
5.1.4	Setting the Topic	15
5.1.5	Moderated Chat	15
5.1.5.1	Moderating	16
5.1.5.2	Special Guests	16
5.1.6	Chat Transcripts	16
5.2	Server-wide Operators	17
5.2.1	Becoming a Server-wide Operator	17
5.2.2	Managing Server-wide Operators	17
5.2.3	Kicking Users Out of the Server	18
5.2.4	Banning Users from the Server	18
5.2.5	Commands for Server-wide Operators Only	18

**Command Index . . . . . 20**

# 1 General Information

## 1.1 Basic Concepts

A chat server is one important component of a chat system. The other one is a chat client, which is usually a Java applet. It communicates with the chat server for posting and receiving messages sent by users. The chat applet can be embedded in a web page and runs in web browsers on all platforms supporting Java applets.

In the case of the POLChat system, there is also a nick server, which stores information about users (received from **POLChat Server**) in a database. However, the nick server is not required for **POLChat Server** to work correctly (though users will not be able to protect their (nick) names with passwords, have buddy lists, etc. without it).

## 1.2 Main Features

- simple to use and install
- small (around 50K without images)
- fast (uses in-the-background initialization)
- fully customizable with applet parameters
- client's source code available (!) for maximum customizability (under the GPL license)
- rich administrative capabilities (room/server operators and half operators, temporary and permanent)
- buddy and ignore lists support
- posted messages can be colored, in different font styles, and contain small images embedded in them
- e-mail and www addresses auto-detected and presented as clickable links
- full Unicode support
- HTTP tunneling support (allows the chat client to work from behind firewalls or with proxy servers)
- moderated chat support
- chat transcripts in HTML format delivered via e-mail

## 2 Installation

**POLChat Client** comes in one `tar.gz` archive, which includes, among some documentation files, a `bin` directory. This directory contains three different versions of the chat applet, each in a separate subdirectory. They are:

1. `std`—a standard, full version with images (of buttons) in separate files (thus easily changeable)
2. `std.gfxemb`—a standard, full version with all images embedded in a class file (thus fast to load)
3. `light`—a non-standard version without images and some of the functionality (no *Preferences* or *Rooms* windows, for example); almost half the size of the `std.gfxemb` version, thus even faster to load

Installing **POLChat Client** on a web page consists of two steps. The first one is copying one of the above directories into a directory accessible from the web with HTTP (it is done only once even if the chat is installed on more than one web page). The second step is to include a reference to the applet on the web page. This reference has the following syntax:

```
<APPLET CODE=POLChat.class ARCHIVE=arch_name CODEBASE=arch_base WIDTH=w HEIGHT=h>
</APPLET>
```

Where *arch\_name* is the name of the archive containing the applet (e.g. the `std` directory has the `c.jar` archive), *arch\_base* is the base URL of this archive (e.g. if `c.jar` is accessible through `http://www.some.domain.com/chat/c.jar`, then the base URL is `http://www.some.domain.com/chat/`), and *w* and *h* specify (in pixels), respectively, the width and height of the area the applet will occupy on the page it is installed.

For example, if the `std` version of **POLChat Client** has been installed in a `chat` directory on host `host.domain.com`, made accessible with HTTP as `http://host.domain.com/chat/c.jar`, and decided to take a 640 by 450 area on a web page, then this page should contain the following reference:

```
<APPLET CODE=POLChat.class ARCHIVE=c.jar CODEBASE=http://host.domain.com/chat/ WIDTH=640 HEIGHT=450>
</APPLET>
```

However, the chat installed in the way described above will not work in older browsers supporting only Java 1.0.2. The JAR archive containing the applet must be extracted (e.g. with `unzip`) in order for the chat classes to be accessible to older browsers.

After the line containing the applet opening tag (`<APPLET CODE=...`) and before the closing tag (`</APPLET>`), there can be various parameters specified to the applet. See [Chapter 3 \[Client Customization\], page 3](#) for the list of available parameters.

**Note that the addresses of web pages containing a reference to the chat applet must be specified in the chat server configuration in order for the applet to be able to connect to the chat server.**

## 3 Client Customization

This chapter lists parameters that can be specified in the HTML file containing a reference to **POLChat Client**. The parameter specification must be included between ‘APPLET’ opening and closing tags, and it has the following format:

```
<param name="parameter_name" value="parameter_value">
```

For parameters of the Boolean type (e.g. those specifying the presence of some components), acceptable values are ‘true’ (‘yes’, ‘1’) and ‘false’ (‘no’, ‘0’).

The example below specifies the *room* parameter value as ‘sample room’, and the *bgcolor* parameter value as ‘#f0f0f0’ (a color code).

```
<APPLET CODE=Chat.class ARCHIVE=c.jar CODEBASE=s1.polchat.com WIDTH=100% HEIGHT=100%
  <param name="room" value="sample room">
  <param name="bgcolor" value="#f0f0f0">
</APPLET>
```

### 3.1 Logging In

The table below contains parameters related to logging the user in.

‘room’	the name of a room the user will join automatically after logging in successfully; it must be the name of an existing (standard) room (listed in the chat server configuration), or else the user will not join any room (if users’ rooms are disallowed in the server configuration) or the room will be created and the user joining it will become its operator (users’ rooms allowed); if unspecified, it defaults to the address of the web page containing the applet; specifying this parameter is <b>highly</b> recommended!
‘nick’	the (nick) name of the user entering the chat (by default, unspecified); see description below
‘pass’	the user’s password (by default, unspecified); see description below
‘cookie’	cookie identifying the user (by default, unspecified and <b>not used</b> ); see description below

If neither the *nick* nor the *cookie* parameter is specified (the default setting), **POLChat Client** presents a login panel in which the user can enter his or her (nick) name and possibly the password (the presence of the password input field is controlled by the *password\_input* parameter; by default, it is not presented; see [Section 3.2 \[Components\], page 4](#)).

If the *nick* or *cookie* parameter is specified, the applet sends *nick*, *pass*, and *cookie* values to the chat server (unspecified values default to empty strings), which in turn sends them to the nick server (if the chat server is not connected to the nick server, it ignores *pass* and *cookie* values and accepts only the user’s name). By default, the nick server ignores the *cookie* value—it only checks the user’s password with the one stored in the database (if information about the user is found in the database), and sends back a response to the chat server. This response contains the user’s name, preferences, buddy list, etc.

The *cookie* parameter can be used in the following way: A registered user is required to log in before entering a page containing the chat applet. The logging in script assigns each user a unique cookie value which is sent to the user's browser and stored in the database. When the user enters a page containing the chat, this cookie is specified to the applet using the *cookie* parameter. The applet sends this cookie to the chat server, which in turn sends it to the nick server. By default, the nick server ignores cookies, but it can be changed (**POLChat Nick Server** comes with full source code under the GPL license) to look in the database for a user with the specified cookie value, and, if it finds one, to send his or her name, preferences, etc. to the chat server.

## 3.2 Components

The table below contains parameters (all of Boolean type) specifying the presence of miscellaneous components (by default, all elements are shown except the password input field).

<code>'password_input'</code>	show the text field for entering the password (in the login panel)
<code>'temporary_name'</code>	show checkbox for specifying whether the name entered in the login panel should be considered temporary (no password needed) or registered (password required)
<code>'nick_list'</code>	show the list of users
<code>'nick_list_toolbar'</code>	show the list of users toolbar with <code>'whisper'</code> , <code>'ignore'</code> , <code>'buddy'</code> , and <code>'info'</code> buttons
<code>'nick_number'</code>	show the status bar containing the number of users in the current room
<code>'room_info'</code>	show the status bar containing the room's name and description
<code>'rooms_button'</code>	
<code>'preferences_button'</code>	show buttons for opening the <i>Rooms</i> and <i>Preferences</i> windows, respectively
<code>'categorized_list'</code>	show categorized list of rooms
<code>'float_button'</code>	show the <code>'float'</code> button
<code>'toolbar_default'</code>	the presence of the toolbar used for selecting colors, styles, etc. of the composed message changes dynamically depending on the configuration of the room in which the user is present; in rooms with plain-text messages, this toolbar is

hidden, while in other rooms, it is shown; by default, when a user logs in, and before he or she enters a plain-text room, this toolbar is shown and hidden immediately after the information about the current room arrives; in order to avoid this seen-for-a-moment-toolbar effect, the toolbar can be hidden by default

### 3.3 URLs

The table below contains parameters specifying miscellaneous URLs.

`'images_url'`

the URL of the web page containing the list of images that can be embedded in messages posted by users using `<image_name>` tags; if this parameter is specified, a button labeled `'images'` will appear in the text toolbar (the one for selecting colors, styles, etc. of composed messages)

`'exit_url'`

the URL to which to transfer the user when he or she quits the chat

`'exit_url_target'`

the target frame in which to open the `exit_url`; can be `'_parent'` (parent frame), `'_top'` (top-level frame), `'_self'` (applet's frame—the default one)

`'profile_url_prefix'`

prefix of a URL for showing user profiles; it is used when a user selects a user name on the users list and presses the `'info'` button (fourth under the list of users); a new browser window pointed at the URL, this being a concatenation of the given URL prefix and the selected user name, is then opened (e.g. if the prefix is specified as `'http://profiles.polchat.com/show.php?user='`, pressing the `'info'` button after selecting a user named `'john'` will then open a new browser window pointed at the `'http://profiles.polchat.com/show.php?user=john'` URL).

`'webcam_url'`

an image from the specified URL will be shown above the users list; **it must be located on the same host as the chat client**; good for celebrity chats

`'webcam_refresh'`

refresh time (in seconds) for the above image; if greater than 0, `'webcam_url'` must end with either `'?'` or `'&'`; this is needed to overcome the cache used by the Java Virtual Machine, which is done by appending subsequent integers to the end of the address

### 3.4 Preferences

The table below contains parameters specifying default values for preferences that can be set by the user in the *Preferences* window.

`'pref_font_name'`

<code>'pref_font_size'</code>	the name ('Courier', 'Helvetica', or 'TimesRoman') and size (10-18) of the font used to display messages posted by users (default: 'Helvetica', 14)
<code>'pref_ignore_colors'</code>	ignore colors in messages posted by users (default: 'no')
<code>'pref_ignore_images'</code>	ignore images in messages posted by users (default: 'no')
<code>'pref_show_traffic'</code>	show information about users joining and leaving the room (default: 'yes')
<code>'pref_beep_incoming'</code>	beep when a new user joins the room (default: 'no')
<code>'pref_ignore_private'</code>	ignore private messages (default: 'no')
<code>'pref_separate_windows'</code>	show private messages in separate windows (default: 'yes')

### 3.5 Colors

The table below contains parameters related to the colors of components.

<code>'bgcolor'</code>	background color (default: '#ffffff')
<code>'bgtextcolor'</code>	background text color (default: '#000000')
<code>'fgcolor'</code>	foreground color (the status bar containing that room's name and description is in this color, for example)
<code>'fgtextcolor'</code>	foreground text color
<code>'color_names'</code>	names of colors that can be embedded in messages posted by users; it has the following format: <code>'name1=#rrggbb, ..., nameN=#rrggbb'</code> ; for example, <code>'black=#000000, red=#ff0000, green=#00cf00, blue=#0000df'</code>

Note that there are no parameters for specifying colors of ignored users, buddies, etc. so that they remain uniform across chats from different sites.

### 3.6 Fonts

The table below contains parameters specifying the names and sizes of fonts used by the applet (acceptable names of fonts are 'Courier', 'Helvetica', and 'TimesRoman').

<code>'login_font_name'</code>
<code>'login_font_size'</code>

`'login_small_font_name'`  
`'login_small_font_size'`  
names and sizes of fonts (normal/small) used in the login panel

`'font_name'`  
`'font_size'`  
`'small_font_name'`  
`'small_font_size'`  
names and sizes of fonts (normal/small) used in components such as buttons, checkboxes, lists, labels, etc.

`'desc_font_name'`  
`'desc_font_size'`  
name and size of the font used to display the room's name and description

`'nick_list_font_name'`  
`'nick_list_font_size'`  
name and size of the font used in the users list

`'room_list_font_name'`  
`'room_list_font_size'`  
name and size of the font used in the rooms list

### 3.7 Connections

The table below contains parameters related to the connection used for communicating with the chat server.

`'port'` the port number to use when connecting to the server (default: 14003)

`'use_tcpip'`  
use TCP/IP connections (default: `'yes'`)

`'use_http'`  
use HTTP (tunneled) connections (default: `'yes'`)

`'http_connection_selection'`  
show the HTTP connection selecting checkbox in the login panel (default: `'no'`, HTTP connection will be tried if TCP/IP fails)

`'http_connection'`  
the HTTP connection selecting checkbox default value (default: `'yes'`)

## 4 Using the Chat

This chapter describes features of the chat applet available to all users.

### 4.1 Entering the Chat

In order to enter the chat, by default the user is required only to provide his or her (nick) name in the login panel. After that, the chat applet tries to establish a connection to the chat server (the server it originates from). If all goes well, that is the connection is established successfully, and the name is OK (no other users by the same name already on the server, no forbidden characters in the name, etc.), the login panel is switched to the main chat panel where the user can post his or her messages, see messages posted by other users, etc.

The login panel may also contain an input field for the password (its presence is controlled by the *password\_input* parameter; see [Section 3.2 \[Components\], page 4](#)). When the chat server is connected to the nick server, users can have their (nick) names registered and available only after providing a valid password. Trying to use a name without providing the required password will result in an unsuccessful login (the user will be informed of the problem with an appropriate message). The chat server may also be configured not to accept users who are not registered (by default, it accepts all users—both registered and unregistered).

It is possible to move the login process outside of the applet onto a web page. For example, the user may be required to enter his or her name and password on one web page, after which the web page with the chat will be presented (see [Section 3.1 \[Logging In\], page 3](#) for details of how to achieve this). A good reason for moving the login process outside of the applet is when integrating the chat with a site where users must log in before using some of its services (one of which could be the chat). Then, a user already logged in will not have to enter his or her name and password for the second time when entering the chat.

### 4.2 Joining a Room

After the successful login, the user automatically enters a room, the name of which is specified to the applet with the *room* parameter (see [Section 3.1 \[Logging In\], page 3](#)).

The user can also join a room manually, by issuing the `/join` command followed by the name of the room to join. For example, the command `/join warsaw` will make the user join the ‘warsaw’ room.

If the user gives the name of a non-existing room to the `/join` command, this room will be created automatically if the chat server is configured to accept users’ rooms, or else the `/join` command will fail. If the room is created, the user joining it becomes its operator.

The user may also join a room (or create one) in the *Rooms* window. See [Section 4.9 \[Rooms Window\], page 12](#) for details.

## 4.3 Posting Messages

There are two kinds of messages users can post, namely, public and private messages. When a public message is posted, it appears almost immediately to everyone present in the same room as the user posting the message. Private messages, in contrast, appear to only one user, the one to whom the message is addressed.

Posting a public message is done by typing it in the text field at the bottom of the main chat panel, and pressing the `ENTER` key.

Private messages can be posted in two ways. One is very similar to posting public messages, but the message must be preceded by the `/msg` command followed by the name of the user to whom the message is addressed (e.g. `/msg john remember me?` will post the message `remember me?` to a user named `john`). The other way is by opening a separate window for exchanging messages with the selected user only (this is done by first selecting the user's name from the list, and then pressing the `whisper` button—first under the list of users), typing the message in the text field at the bottom of the opened window, and pressing the `ENTER` key. Unless the `show private messages in separate windows` option is not checked in the *Preferences* window (See [Section 4.8 \[Preferences Window\]](#), page 11), the window will stay open allowing convenient way of exchanging private messages with the selected user without the need to precede each message with the additional `/msg name` command.

There can be multiple private chat windows open at a time. These windows also open automatically when private messages from new users arrive (a new window is opened if there is currently no window for exchanging private messages with the user who posted the new message). If the `show private messages in separate windows` option is not selected, all private messages appear in the main chat panel.

Both public and private messages are not anonymous, because they are preceded with the name of the user who posted them. In the main chat panel, private messages can be distinguished from public ones by characters surrounding the user's name—by default, asterisk characters (`*`) surround names in private messages (e.g. `*joan* john, is that you?` is a private message `john, is that you?` from a user named `joan`). However, these characters can be changed to different ones in the chat server configuration.

By default, the text of messages posted by users can be colored, as well as in different font styles (bold, italic, underlined). The toolbar under the text field for posting messages can be used for changing the color and style of the message being composed. Its use is similar to those of popular word processors.

By default, small images can also be embedded in messages. Embedding an image is done by simply writing the image name in angle brackets (e.g. `<happy>` embeds an image named `happy`). By default, there are only two images available, but more can easily be added by the chat server administrator (see *POLChat Server Reference Manual* for details). The chat server administrator should provide users with the full list of available image names, for example, by creating a web page with images and their names. Moreover, a special parameter can be used to create a button entitled `images`; pressing it will cause a new browser window to be opened at the specified address; see [Section 3.3 \[URLs\]](#), page 5).

Note that the features described above (colors, styles, images) may not be available in some rooms, because rooms can be configured to allow plain text messages only.

## 4.4 Ignoring Users

A user may wish to ignore (to not see) public and private messages posted by selected users. To do this, the `/ignore` command followed by the name of the user to ignore can be used (e.g. `/ignore john` will make messages posted by a user named `john` not appear to the user issuing the command).

The `/unignore` command can be used to reverse the effect of the `/ignore` command. While the `/ignore` command adds a user to the ignore list, the `/unignore` command removes a user from it (e.g. `/unignore john` will remove a user named `john` from the ignore list; if there is no user named `john` in the ignore list, this command will have no effect when issued).

A more convenient way of adding and removing users to the ignore list is to select a user name on the list of users and press the `ignore` button (second under the list of users). If the selected user name is not on the ignore list, it will be added to it, and if the user's name is already there, it will be removed. However, this method is available only if the list of users is presented—its presence may be turned off in the chat server configuration or with the `nick.list` parameter (see [Section 3.2 \[Components\], page 4](#)).

For displaying the ignore list, the `/ignored` command can be used. It lists all users ignored by the user issuing the command. Ignored users present in the same room as the user ignoring them are marked on the list of users with the gray color.

Ignore lists are saved between chat sessions if the chat server is connected to the nick server. Only lists of registered users are saved.

## 4.5 Buddies

In a room containing 500 users, it may be hard for a user to decide which of his or her friends are online just by glancing at the list of users. To cope with this problem, users can have buddy lists. Any user may be added to another user's buddy list with the `/buddy` command followed by the name of the user to add (e.g. `/buddy joe` will add a user named `joe` to the buddy list of the user issuing the command). Since names of buddies are marked on the list of users with the green color, this makes finding which friends are online (in crowded rooms) simple and fast!

Just like with the ignore list, there is a way to remove a user from the buddy list. The `/unbuddy` command followed by the name of a user will remove him or her from the buddy list of the user issuing the command.

The results of the two commands described above can also be achieved by selecting a user on the list of users, and pressing the `buddy` button (third under the list of users). This will cause the selected user to be added to (removed from) the buddy list if he or she is not (already is) on it.

The `/buddies` command is similar to the `/ignored` one—it lists names of users, in this case, those included on the buddy list. There is, however, one more difference between these two commands. On the list displayed by the `/buddies` command, the name of a user is followed by the name of the room in which he or she is present at the moment (if any).

Buddy lists are saved between chat sessions if the chat server is connected to the nick server. Only lists of registered users are saved.

## 4.6 Quitting the Chat

Quitting the chat may be done either by pressing the ‘quit’ button or by issuing the `/quit` command optionally followed by the farewell message (e.g. `/quit so long!`). The farewell message (if any) will be seen by other users along with the standard message informing about the user’s departure.

When a user quits the chat, the login panel is shown, and the user can immediately re-enter the chat unless the login process has been moved outside of the applet (see [Section 4.1 \[Entering the Chat\], page 8](#)). The user may also be transferred to a specified web page (i.e. the web page with the chat will be changed for another one, the address of which can be specified with the `exit_url` parameter; see [Section 3.3 \[URLs\], page 5](#)).

A user may also quit a room without quitting the chat. The `/part` command can be used for that. The user will be present in no room and posting public messages will not be possible after issuing this command. The user will be able to re-enter any room with the `/join` command (see [Section 4.2 \[Joining a Room\], page 8](#)).

## 4.7 Miscellaneous Commands

The `/topic` command can be used to display the current topic of the room in which the user is present at the moment of issuing the command. The topic is also displayed automatically after entering a room (unless there is no topic), as well as when it is changed by the room operator (see [Chapter 5 \[Using the Chat as an Operator\], page 13](#)).

The `/info` command followed by a user’s name displays brief information about the specified user, including the room name he or she is present in at that moment, time spent online, time for which the user has been idle, etc. If the user identified by the specified name is not logged in, an appropriate message is displayed. Just like ignoring a user selected on the list of users, or adding him or her to the buddy list, information about a user can also be displayed by pressing a button, the ‘info’ button in this case (fourth under the list of users). Pressing this button may also open a new browser window with a profile page of the selected user—this happens if the `profile_url_prefix` parameter is specified to the applet (see [Section 3.3 \[URLs\], page 5](#)).

The `/me` command followed by a message causes this message to be displayed with the user name not followed by the name-message separator (by default, the colon is used; it may be changed to any other character or a few characters in the chat server configuration). This command is commonly used to express emotional states. For example, if the user’s name is ‘john’, the command is `/me gets bored...`, then ‘john gets bored...’ will appear to all users (typing just ‘gets bored...’ will result in ‘john: gets bored...’).

## 4.8 Preferences Window

Users can customize several chat features as they see fit. Pressing the ‘preferences’ button (its presence may be disabled—see [Section 3.2 \[Components\], page 4](#)) shows the *Preferences* window with options regarding font types, font sizes, messages, beeping, etc.

The *Preferences* window may also contain a ‘protect with password’ checkbox and a text field for the password (its presence is dependent on the server configuration—it must be connected to the nick server and allow registration inside the applet; see *POLChat Server Reference Manual* for details). Checking this checkbox and entering a password makes the user’s name available only after providing this password when logging in.

Just like buddy and ignore lists, preferences are saved between chat sessions for registered users.

## 4.9 Rooms Window

Near to the ‘preferences’ button, there can be a ‘rooms’ button (its presence may be disabled—see [Section 3.2 \[Components\], page 4](#)). Pressing it shows the *Rooms* window. In it, the user can join another room either by selecting it from the list of available rooms (**only public rooms are listed**) or entering its name.

Unless the chat server is configured not to allow users’ rooms, there is also a panel for creating users’ own rooms. A short description of the room, whether it is public or private, and the room category (if there are any in the chat server configuration) can be specified, along with the name of the room to create.

## 5 Using the Chat as an Operator

Not all features of the chat system are available to all of its users. Some are reserved for a special kind of privileged user only, namely, the *operator*. An operator can, among other things, kick selected users out of the room (or the server), moderate discussions, get the public chat transcript, change the room topic, etc.

There are two categories of operators: *room operators*, the rights of whom are limited to a single room, and *server-wide operators*, the room operator rights of whom are unlimited (i.e. they have them in all rooms), and who have some server-oriented rights (like kicking users out of the server, not just the room).

There is also a special kind of limited operator named *half operator*, the rights of whom can be fully customized in the chat server configuration except for the right to give (take away) any operator rights to (from) other users, which is disallowed.

Below is a list of what can be globally allowed or disallowed for half operators:

- kicking users out of the room (and the server for server-wide operators)
- banning users from the room (the server)
- changing the room topic
- moderating discussions
- selecting special guests (users whose messages do not need to be moderated before they appear to other users)
- getting the public chat transcript

Moreover, the rights of half operators can be either permanent or temporary. Permanent rights are only available when the chat server is connected to the nick server—they are then stored along with other information about registered users. Temporary half operator rights remain until the half operator leaves the room (in the case of room half operators) or the server (server-wide half operators).

The names of operators and half operators are distinguished on the list of users with a special color (by default, red).

### 5.1 Room Operators

Except the *Managing Room Operators* subsection, all features discussed in this section are available to both room operators and half operators. However, the rights to certain commands may be denied for half operators in the chat server configuration.

#### 5.1.1 Managing Room Operators

Any room operator (not a half operator) can give his or her operator rights to other users. If the chat server is connected to the nick server, the user being given the rights does not need to be present in the room for which the operator rights are given.

In order to give operator rights to a user, the `/op` command followed by the user's name can be used. For example, if a user named 'john' is an operator, he can issue the `/op joan` command to give operator rights to a user named 'joan'.

For giving half operator rights, the `/halfop` command followed by the name of the user to whom to (permanently) give the rights can be used.

The `/tmphalfop` command is similar to the `/halfop` one except it gives **temporary** half operator rights to the specified user.

Operator rights can also be taken away from any user by any operator using the `/unop` command followed by the name of the user from whom to take the rights away (permanently).

The `/ops` command can be used to list the names of users with **permanent** room operator and/or half operator rights. Note that the list will always be empty if the chat server is not connected to the nick server. Also, in such case, all commands related to giving permanent operator rights give users temporary rights instead. This is the only case when a temporary operator can appear—there cannot be temporary operators when the chat server is connected to the nick server, only temporary half operators.

### 5.1.2 Kicking Users Out of the Room

There are a few commands that allow operators to kick selected users out of the room. The `/kick` command (in short `/k`) followed by a user's name causes the selected user to be kicked out of the room. Additionally, the reason for kicking the user out of the room may be specified after the user's name. For example, the `/kick john advertising other chats` command will kick a user named 'john' out of the room, and display him the reason for being kicked out ('advertising other chats'). The `/kick john` command will do the same, but the user will not be shown any message explaining the reason for being kicked out. The same result may also be achieved by selecting the user's name on the list of users, and pressing the `/kick` button. This button, along with the `/ban` one, is shown to all operators instead of the status bar with the number of users in the room.

When specifying user names, use of the asterisk character ('\*') as the first or the last character of the name is allowed, in which it denotes an arbitrary string of characters (including empty). For example, the `/kick jo*` command will kick all users, the names of whom start with 'jo' (e.g. 'john', 'jo', 'JohnTheHacker'). Note that the names of users are not case sensitive. Moreover, when using the asterisk character in the name specification, the reason for being kicked out cannot be given.

The `/kickip` command is similar to the `/kick` one except it accepts an IP address as its argument, and causes all users connected from the specified address to be kicked out of the room. (Operators can see IP addresses of incoming users unless this feature is disabled in the chat server configuration—privacy concerns may be one reason for doing so.) For example, the `/kickip 127.0.0.1` command will kick all users connected from '127.0.0.1' out of the room. When specifying the address, it is also possible to use the asterisk character for denoting arbitrary number in the range of 0-255. For example, '127.0.\*.\*' denotes all IP addresses starting with '127.0'—the last two numbers can be anything in the range of 0-255.

It is also possible to kick all users connected from the IP addresses of selected users out of the room. The `/kickipof` command followed by a user name can be used for it. For example, the `'/kickipof john'` will kick all users connected from the same IP address as that of a user named `'john'`.

### 5.1.3 Banning Users from the Room

When a user is kicked out of the room, he or she can easily return to it using the `/join` command. Fortunately, it is possible to deny access to the room from selected IP addresses for a period of time. The commands which can be used for it are `/ban` (in short `/b`), `/banip`, and `/banipof`. They are similar to, respectively, `/kick`, `/kickip`, and `/kickipof`, that is they cause selected users to be kicked out of the room, plus they deny access from the IP addresses in question (specified either explicitly, or implicitly with user names) for a period of time. The reason for banning a user from the room may also be given as the last argument of the `/ban` command.

The time (in minutes) for which the access is denied can be specified as the second argument of the banning commands; if not specified, the default ban time is used. Users can set the default ban time using the `/bantime` command followed by the time to be set as default. If no time is specified, the `/bantime` command displays the current value of the default ban time.

Examples:

- `'/ban john 120 spamming'` will kick a user named `'john'` out of the room, deny access from his IP address for two hours (120 minutes), and present him the reason for being banned from the room (`'spamming'`)
- `'/banip 127.0.0.* 60'` will kick all users connected from IP addresses starting with `'127.0.0.'` out of the room, and deny access from `'127.0.0.*'` for one hour (60 minutes)
- `'/ban john'` issued after `'/bantime 180'` and before any other `'/bantime'` command, would kick a user named `'john'` out of the room, and deny access from his IP address for the default ban time, that is three hours (180 minutes)

Banned IP addresses can be managed with two commands, namely, `/bans` and `/unban`. The first one lists banned IP addresses along with the time remaining for ban's expiration. The `/unban` command followed by an IP address removes the specified address from the list of IP addresses, the access from which is denied.

Selecting a user's name from the list of users, and pressing the `'/ban'` (presented to all operators above the list of users) button has the same effect as using the `'/ban'` command followed by the user's name.

### 5.1.4 Setting the Topic

Any room operator can change the current room topic using the `/topic` command followed by a new topic. For example, the `'/topic It's party time! :)'` will change the current room topic to `'It's party time! :)'`.

In the topic, the use of font, color, and image tags is allowed, just like in ordinary chat messages (e.g. `'<b>this is in bold..'`).

## 5.1.5 Moderated Chat

A moderated chat is one in which messages posted by users must first be accepted by the room moderator (see [Section 5.1.5.1 \[Moderating\], page 16](#)) before they appear to other users. However, any room operator can select users who can post messages without prior moderation. These users are called *special guests* and they are distinguished on the list of users with a special color (by default, blue).

### 5.1.5.1 Moderating

Any room operator can issue the `/moderate` command to make the room a moderated one. This command either fails, if the room is already moderated by another operator (an appropriate message will appear), or succeeds, making the user who issued the command a moderator, and showing him or her a special moderation panel with two message queues.

The first queue contains all messages posted by users, each preceded with three HTML-like links: `show`, `enqueue`, and `delete`, clicking on which causes the following message to be, respectively, shown to other users, enqueued in the second queue, or deleted. Note that messages are deleted from the queue even when they are shown to other users or enqueued in the second queue, not just when deleted explicitly (it would not make sense to click a link to show a message to other users, and then click again to delete it).

Messages contained in the second queue are preceded with two HTML-like links: `show` (for showing the message to other users) and `delete` (for deleting it). This queue can be used by the moderator, for example, for enqueueing interesting messages for later use.

A moderated chat stops when the moderator leaves the room, or when he or she issues the `/moderate stop` command. This command is issued automatically when the moderation panel is closed.

### 5.1.5.2 Special Guests

Room operators can select users (including themselves), the messages of whom do not need to be moderated before they appear to other users during the moderated chat. These users are called *special guests*, and the `/guest` command followed by a user's name can be used for making the specified user a special guest. For example, `/guest john` will make a user named `'john'` a special guest.

Special guests are distinguished on the list of users with a special color. By default, it is blue, but more than one color can be used (the full list of possible colors is defined in the chat server configuration). The `/guest` command accepts a second argument which is a color number to use for the specified user. For example, `/guest john 2` will make a user named `'john'` a special guest and display his name on the list of users with the color number two (the value of which is defined in the chat server configuration).

The special guest title may also be taken away from a user. The `/unguest` command can be used for it, and it should be followed by the name of the special guest user from whom to take the title away.

Note that special guests can also exist during a non-moderated chat.

### 5.1.6 Chat Transcripts

It is possible to have a transcript of messages posted by users during a period of time (this is useful for celebrity chats, for example).

The `/transcript` command is used for managing the transcript. When issued without parameters, it displays general information, like whether transcript logging is turned ON or OFF, the current and maximum number of messages in the transcript queue, etc. The following parameters are acceptable by the `/transcript` command:

- `start`—turns the transcript logging ON; from now on, all messages posted by users are logged (saved) in the transcript queue
- `stop`—turns the transcript logging OFF
- `reset`—clears the transcript queue
- `send`—sends the transcript to the e-mail address specified as the third argument; the transcript is sent in the HTML format with all messages' colors, font styles, etc. preserved (e.g. `/transcript send john@guesswhat.com` will send the transcript to `'john@guesswhat.com'`)

When the maximum number of messages in the transcript queue is reached, posting new messages causes the oldest one to be removed. The maximum number of messages in the transcript queue can be changed in the chat server configuration.

## 5.2 Server-wide Operators

Server-wide operators have room operator rights in all rooms, as well as the privilege to kick users out and ban from the chat server (see [Section 5.2.3 \[Kicking Users Out of the Server\]](#), page 18, and [Section 5.2.4 \[Banning Users from the Server\]](#), page 18). Accordingly, server-wide half operators have half operator rights in all rooms.

Server-wide operators (but not half operators) may also use two special commands, namely, `/push` and `/broadcast` (see [Section 5.2.5 \[Commands for Server-wide Operators Only\]](#), page 18).

### 5.2.1 Becoming a Server-wide Operator

A user becomes a server-wide operator automatically after entering the chat server (not necessarily a room) if he or she has earlier been given permanent server-wide operator rights, or may become one manually using the `/loginop` command. This command accepts one argument which is the login name of a server-wide operator defined in the chat server configuration (in `'oppasswd'` file—see *POLChat Server Reference Manual* for details). After that, a password input field appears; entering a valid password in it makes the user a server-wide operator.

For example, if the ‘`oppasswd`’ configuration file (on the server) contains a user ‘`polchat`’ with password ‘`abc123`’, issuing the ‘`/loginop polchat`’ command, and entering password ‘`abc123`’ afterwards makes the user issuing the command a server-wide operator.

Note that in the chat server configuration, there can be restrictions for IP addresses from which the use of the `/loginop` command is allowed.

## 5.2.2 Managing Server-wide Operators

The commands for managing server-wide operators are similar to those for managing room operators except they are preceded with ‘`server`’, or in short ‘`s`’ (see [Section 5.1.1 \[Managing Room Operators\]](#), page 13). They are:

- `/serverop` (or `/sop`) followed by a user’s name makes the specified user a server-wide operator (permanent)
- `/serverhalfop` (or `/shalfop`) followed by a user’s name make the specified user a server-wide half operator (permanent)
- `/servertmphalfop` (or `/stmphalfop`)—similar to `/serverhalfop` (`/shalfop`) except the specified user is made a **temporary** server-wide operator
- `/serverunop` (or `/sunop`) followed by a user’s name takes away the server-wide operator and half operator rights from the specified user
- `/serverops` (or `/sops`) lists names of users who have permanent server-wide operator or half operator rights (the list will be empty if the chat server is not connected to the nick server)

## 5.2.3 Kicking Users Out of the Server

Just like kicking users out of the room (see [Section 5.1.2 \[Kicking Users Out of the Room\]](#), page 14), it is possible to kick a user out of the chat server. The commands which can be used for it are `/serverkick` (in short `/skick`, even shorter `/sk`), `/serverkickip` (`/skickip`), and `/serverkickipof` (`/skickipof`). They accept the same arguments as `/kick`, `/kickip`, and `/kickipof`, respectively, and work in the same way except that users are kicked out of the server, not just the room.

## 5.2.4 Banning Users from the Server

Just like banning users from the room (see [Section 5.1.3 \[Banning Users from the Room\]](#), page 15), it is possible to ban users from the chat server (banned are actually IP addresses, not users). The commands which can be used for it are `/serverban` (in short `/sban`, even shorter `/sb`), `/serverbanip` (`/sbanip`), and `/serverbanipof` (`/sbanipof`). They accept the same arguments as `/ban`, `/banip`, and `/banipof`, respectively, and work in the same way except that users are kicked out of the server, and IP addresses are banned from the server, not the room.

The two commands for managing IP addresses banned from the room, namely, `/bans` and `/unban` (for showing the list of banned IP addresses, and removing from this list, respectively) also have their server-oriented equivalents. They are `/serverbans` (in short `/sbans`) and `/serverunban` (`/sunban`), and they work in the same way as `/bans` and `/unban`, respectively, except they affect the list of IP addresses banned from the server, not the room.

### 5.2.5 Commands for Server-wide Operators Only

There are two commands that can be used only by server-wide operators (server-wide half operators cannot use them).

The first one is `/push`. It accepts three parameters: the name of a user (the use of the asterisk character as the first or the last character is permitted), a web address, and a target argument. The `/push` command causes the selected web address to be opened in a web browser window for selected users. The target can be either the name of a browser window to use (successive pushes in the same window will replace its content) or a special name starting with the underscore character (`'_'`), like `'_blank'` (for a new, unnamed window), `'_self'` (for the window and frame that contains the applet), `'_parent'` (parent frame), and `'_top'` (top-level frame). If the target argument is not specified, the `'_blank'` target is used. For example, the `/push as* http://www.guesswhat.com` command will open a new, unnamed web browser window pointed at `'http://www.guesswhat.com'`, for all users, the names of whom start with `'as'`.

The second command is `/broadcast`, and it is used for broadcasting a message to all users (not just to those in the current room). For example, the `/broadcast the chat server is going down in two minutes!!` will show the message `'the chat server is going down in two minutes!!'` to all users connected to the chat server.

## Command Index

/		/sbanipof .....	18
/b .....	15	/sbans .....	18
/ban .....	15	/serverban .....	18
/banip .....	15	/serverbanip .....	18
/banipof .....	15	/serverbanipof .....	18
/bans .....	15	/serverbans .....	18
/bantime .....	15	/serverhalfop .....	17
/broadcast .....	18	/serverkick .....	18
/buddies .....	10	/serverkickip .....	18
/buddy .....	10	/serverkickipof .....	18
/guest .....	16	/serverop .....	17
/halfop .....	13	/serverops .....	17
/ignore .....	10	/servertmphalfop .....	17
/ignored .....	10	/serverunban .....	18
/info .....	11	/serverunop .....	17
/join .....	8	/shalfop .....	17
/k .....	14	/sk .....	18
/kick .....	14	/skick .....	18
/kickip .....	14	/skickip .....	18
/kickipof .....	14	/skickipof .....	18
/loginop .....	17	/sop .....	17
/me .....	11	/sops .....	17
/moderate .....	16	/stmphalfop .....	17
/moderate stop .....	16	/sunban .....	18
/msg .....	9	/sunop .....	17
/op .....	13	/tmphalfop .....	13
/ops .....	13	/topic .....	11, 15
/part .....	11	/transcript .....	16
/push .....	18	/unban .....	15
/quit .....	11	/unbuddy .....	10
/sb .....	18	/unguest .....	16
/sban .....	18	/unignore .....	10
/sbanip .....	18	/unop .....	13