Ərk Commands

Ərk IRC Client Commands Documentation

https://github.com/nutjob-laboratories/erk
https://github.com/nutjob-laboratories/erk-plugins

Version 0.831

Summary

Much like other popular IRC clients, users can input commands into the chat input widget to control the Ərk IRC client. To use these commands, type the command, followed by any arguments, into the same widget that you enter chat text, and press the enter button.

To execute multiple commands at the same time, Ərk also features scripts. Multiple commands (one per line) can be entered into a text file, and Ərk will execute the commands in that file. Scripts can either be executed directly by using the **/script** command, or can be executed automatically on connection to a specific server.

Connection Scripts and “Callable” Scripts

Ərk uses two different kinds of scripts: connection scripts (which can be automatically executed upon connection to a server, and are set in the connection dialog) and “callable” scripts (scripts that are executed by the **/script** command or the “Run Script” button on non-chat windows).

Connection scripts are executed with *no* arguments, and, thus, can’t use the **/argcount** command (if this command is called, it is simply ignored). Other than that, they function just like other scripts.

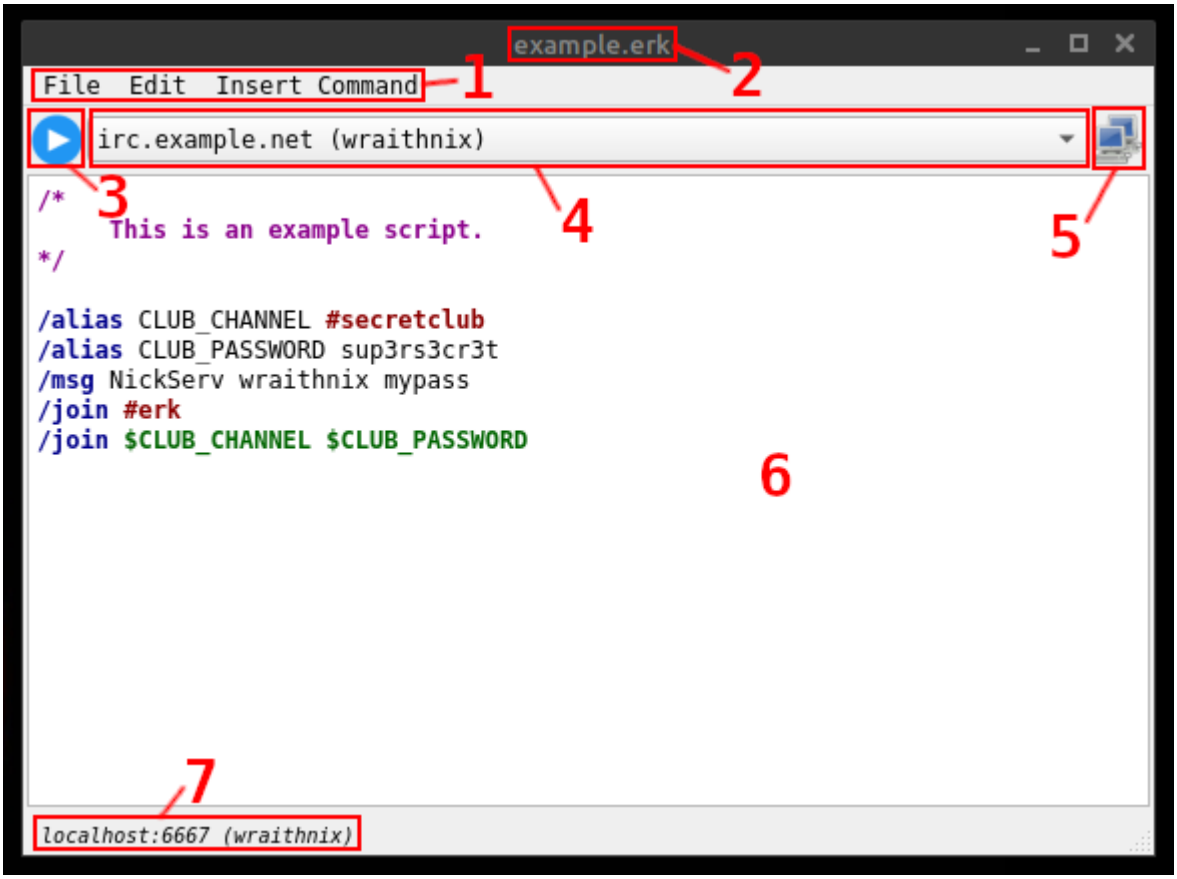
“Callable” scripts are executed with whatever arguments are passed to the **/script** command; scripts executed with the “Run Script” button are executed with no arguments (calls to **/argcount** *will* be executed, however, any script using the command to look for anything other than zero (0) arguments will stop execution).

Ərk scripts have the “.erk” file extension. When using the **/script** command, the file extension does *not* have to be included; when looking for the file, Ərk will append the file extension if necessary.

Ərk will always execute the commands on the server associated with the window that called the script (and connection scripts will always execute on the server being connected to); to execute scripts on any server the client is connected to, use the **Script Editor**, accessible with the **/edit** command.

Script Editor

Ərk features a built-in script editor. It can open, edit, and save scripts, execute them on any server Ərk is connected to, and features syntax highlighting. To open the editor, use the **/edit** command.



1. **Menu bar.** Here you can select scripts to open, save any new scripts or edits, and the normal tools you'd expect to find in a text editor. The "Insert Command" menu contains shortcuts to insert commonly used commands.
2. **File name**. The name of the script being edited.
3. **Execute.** Clicking this button executes the script in the editor on the selected Ərk client connection.
4. **Client selector.** This contains a list of all the servers that Ərk is connected to, allowing you to select which connection you want to execute the script on. The hostname of the server, followed by the nickname in use on that server, is displayed.
5. **Open connection script.** This will load the connection script for the currently selected client. If one does not exist, the editor will be set up to edit and save a new one.
6. **Editor.** Here, you can type your script, or edit any script you open. It features syntax highlighting (comments, commands, and channel names are colored differently from the rest of the text). Currently, there is no option to customize the colors used; this will change in future versions.
7. **Client details.** This displays the hostname/IP address used to connect to the currently selected client, as well as the nickname in use.

IRC Commands

The majority of the available commands are for controlling the IRC connection (for example, joining and leaving channels, sending chat messages, etc). Three commands (**/invite**, **/part** and **/topic**) will behave differently depending on where the command is entered; it will behave one way if it is entered into a chat window (for private chat or channel chat) or if it's entered into a console window (the window which displays data sent by the connected IRC server). One command, **/me**, cannot be entered into a non-chat window. These commands should be familiar to users of other IRC clients such as mIRC or HexChat.

Called from any window
/away [MESSAGE...]
Sets the client to AWAY. If MESSAGE is included, MESSAGE will be sent to any user that sends a private message to the client.

Called from any window
/back
Sets the client to BACK, and disables /away functionality.

Called from a chat window	Called from any window
/invite [CHANNEL] NICKNAME	**/invite CHANNEL NICKNAME**
Invites NICKNAME to the current channel if CHANNEL is omitted; invites NICKNAME to CHANNEL if a channel is included.	Invites NICKNAME to CHANNEL.

Called from any window
/join CHANNEL [KEY]
Joins CHANNEL. If KEY is included, it will be passed to the server as part of the join request.

Called from any window
/list [TERMS]
Displays a list of all channels on the server if TERMS is omitted; displays a list of channels matching TERMS if TERMS is included. TERMS can contain wildcards such as * or ?.

Called from a chat window	Called from any window
/me TEXT	
Sends TEXT to the current chat as a CTCP action message. The command can *only* be called from a chat window.	

Called from any window

/msg TARGET MESSAGE...

Sends a private message to TARGET (which can be a channel or a nickname).

Called from any window

/nick NICKNAME

Changes NICKNAME. If someone is already using NICKNAME, an error is displayed to the client.

Called from any window

/notice TARGET MESSAGE...

Sends a NOTICE to TARGET (which can be a channel or a nickname).

Called from any window

/oper USERNAME PASSWORD

Logs into a server operator account using USERNAME and PASSWORD.

Called from a chat window

/part [CHANNEL] [MESSAGE]

Leaves the current channel if CHANNEL is omitted; leaves CHANNEL if a channel is included. MESSAGE is the message displayed to other users when leaving; MESSAGE is optional.

Called from any window

/part CHANNEL [MESSAGE]

Leaves CHANNEL. MESSAGE is the message displayed to other users when leaving; MESSAGE is optional.

Called from any window

/quit [MESSAGE...]

Disconnects from the current server. If MESSAGE is included, this will be sent to any channels the client is in before it disconnects.

Called from any window

/send MESSAGE

Sends MESSAGE to the server as a raw text; the outgoing message will not be altered. This is to allow the client to send messages to the server that the client doesn't normally support.

Called from any window

/time [SERVER]

Displays local time for the server currently connected to, or the SERVER specified.

Called from a chat window	Called from any window
/topic [CHANNEL] TEXT... Sets the topic for the current channel if CHANNEL is omitted; sets the topic for another channel if CHANNEL is included.	**/topic CHANNEL TEXT...** Sets the topic for CHANNEL.

Called from any window
/version SERVER [SERVER...] Requests server software version information from one or more SERVERs.

Called from any window
/who TEXT Displays a list of users who’s nickname matches TEXT.

Called from any window
/whois NICKNAME Displays information about a specific user.

Called from any window
/whowas NICKNAME [MAXIMUM ENTRIES] [SERVER] Displays information about users with a NICKNAME that no longer exists.

Scripting Commands

The rest of the commands control the Ərk client software itself. All of these commands can be entered into *any* window; however, the **/help** command will display slightly different output depending on what kind of window the command is called from. Four commands, **/argcount**, **/alias**, **/_alias**, and **/wait**, can't be called by any window, and can only be called from scripts. If called with no arguments, the **/connect**, **/reconnect**, **/ressl**, and **/ssl** commands will open the "Connect" dialog with the appropriate options pre-selected.

If a "script only" command is called with the incorrect number of arguments or an argument in incorrect type (for example, calling **/wait** with a non-number argument), script execution will stop immediately and an error message with the reason for the error will be displayed.

Called only from a script

/alias NAME MESSAGE...

Creates a new alias (called an "alias variable" in this document). Any script or command ran after this one (including the commands that follow the **/alias** command) that contains a dollar sign followed by NAME ($NAME) will have that instance replaced with MESSAGE. If the interpolation symbol ($) is included in NAME, it will be stripped, allowing for use of the alias variable normally (thus, if the command **/alias $EXAMPLE My message** is executed, using the alias variable would referenced with **$EXAMPLE**, not **$$EXAMPLE**). This command is *only* available for use in scripts; however, alias variables are interpolated into *any* input, including the text typed into the text input widget. Alias variables are "global"; once created, they are usable in any script or command input. Alias variables can also be overloaded (or overwritten) by any other script.

Called only from a script

/_alias NAME MESSAGE...

Creates a new alias (called an "alias variable" in this document). Unlike the **/alias** command, this alias variable is *not* exported for use in other scripts/commands (thus, it is not "global"); this alias will only be interpolated for the script that it is defined it.

Called only from a script

/argcount NUMBER MESSAGE...

Checks to make sure that a script is called with the proper NUMBER of arguments; if not enough or too many arguments are passed to the script, MESSAGE is displayed and script execution stops immediately. This command cannot be called from connection scripts.

Called from any window

/connect [SERVER] [PORT]

Causes the client to connect to an IRC server. If PORT is omitted, a default of 6667 is used. If SERVER and PORT is omitted, the connection dialog is displayed. The client's current nickname, username, and real name is used.

Called from any window

/connectscript SERVER [PORT]

Opens the connect script for SERVER and PORT and executes it. If PORT is excluded, a default port of 6667 is assumed. Alternately, the SERVER and PORT can be passed to this command in the "SERVER:PORT" format.

Called from any window

/edit [FILENAME]

Opens up the built-in script editor; if a FILENAME is passed as an argument, the FILENAME is loaded into the script editor. If a complete path to FILENAME is not provided, the client will look in the scripts directory (in **settings\scripts**, or set by the -**scripts** command-line flag) for the file; the "**.erk**" file extension can be omitted.

Called from any window

/exit

Disconnects from all servers and exits the client.

Called from a chat window

/help

Displays a list of commands useful in channel and private message sessions, as well as all macros.

Called from any window

/help

Displays a list of all commands and macros.

Called from any window

/macro NAME ARG_COUNT MESSAGE...

Creates a macro[1], allowing for new, custom command creation. NAME is the command that will trigger the command (in the form **/NAME**), ARG_COUNT is the number of arguments that the macro will accept, and MESSAGE is the output of the macro, which will be processed like a command. If ARG_COUNT is set to ***** (asterix), the macro will accept any number of arguments (including zero). If a macro is called with an incorrect number of arguments, an error is displayed, and the macro will not run.

Arguments passed to the macro are interpolated into MESSAGE much like arguments to the **/script** command: instances of **$1** will be replaced with the first argument, **$2**

1 "A macro...is a rule or pattern that specifies how a certain input should be mapped to a replacement output." *Wikipedia*

with the second, and so on. There are other optional symbols that can be interpolated into MESSAGE:

SYMBOL	REPLACED WITH
$0	All arguments
$+	All arguments except for the first argument
$NICK	The nickname the client is using
$USERNAME	The username the client is using
$REALNAME	The real name the client is using
$HOSTNAME	The hostname of the server the client is connected to; otherwise, the server and port, delimited by a colon.
$SERVER	The IP address or hostname used to connect to the server the client is connected to
$PORT	The port address on the server used to connect to the server the client is connected to
$WHERE	If the window the macro is called from is a server console, the hostname or server and port of the server. If the window the macro is called from is a channel or private chat, the name of the channel or the user being chatted with

For example, many IRC clients have included a macro called **/trout**, which, when called with a nickname as an argument, sends a CTCP action message to the current chat describing the user "hitting" the targeted nickname with a trout. To create this macro, you would issue this command:

/macro trout 1 /me slaps $1 with a trout

To call this macro (and "slap" your friend Bob with said trout), you could issue the command **/trout Bob**, which would send the command **/me slaps Bob with a trout**.

Called from any window

/macrohelp NAME MESSAGE...

Sets the text displayed for the NAMEd macro when the **/help** command is used.

Called from any window

/preferences

Opens the "Preferences" dialog.

Called from any window

/print MESSAGE...

Displays MESSAGE in the client's current window. If **/print** is called from a window, the MESSAGE will be displayed in that window; if **/print** is called by any other method (like via the "Run Script" button, or from the script editor), the MESSAGE will be displayed in the server's console window.

Called from any window

/reconnect [SERVER] [PORT]

Causes the client to connect to an IRC server, with the option to reconnect upon disconnection turned on.. If PORT is omitted, a default of 6667 is used. If SERVER and PORT is omitted, the connection dialog is displayed. The client's current nickname, username, and real name is used.

Called from any window

/refresh

Requests a new list of channels from the server, and stores the new list in an internal cache.

Called from any window

/ressl [SERVER] [PORT]

Causes the client to connect to an IRC server via SSL/TLS, with the option to reconnect upon disconnection turned on.. If PORT is omitted, a default of 6697 is used. If SERVER and PORT is omitted, the connection dialog is displayed. The client's current nickname, username, and real name is used.

Called from any window

/script FILENAME [ARGUMENT ...]

Opens FILENAME, reads its contents into memory, and executes it as a list of commands; each additional ARGUMENT, is passed to the script. Any errors found in the script are displayed to the current window. ARGUMENTs can be used in the script much like **/alias** variables: **$1** is replaced with the first argument, **$2** with the second, and so on. To interpolate all ARGUMENTs as a single string, use **$0**. ARGUMENTs are interpolated into the script before **/alias** variables. If a line in a script doesn't contain a valid command, it will be sent to the current window as chat text, just as if the user entered it as chat text; if the current window is a non-chat window, an error will be displayed rather than the text being sent to the server. If a complete path to FILENAME is not provided, the client will look in the scripts directory (in **settings\scripts**, or set by the -**scripts** command-line flag) for the file; the "**.erk**" file extension can be omitted.

Called from any window

/settings

Opens the client's settings dialog.

Called from any window

/ssl [SERVER] [PORT]

Causes the client to connect to an IRC server via SSL/TLS. If PORT is omitted, a default of 6697 is used. If SERVER and PORT is omitted, the connection dialog is displayed. The client's current nickname, username, and real name is used.

Called from any window

/style FILENAME

Loads a style file (created with the “Style Editor”, accessible from the “Preferences” dialog) into the current chat. If a complete path to FILENAME is not provided, the client will look in the scripts directory (in **settings\scripts**, or set by the -**scripts** command-line flag) for the file.

Called from any window

/switch [WINDOW NAME]

Switches the current displayed channel. If WINDOW NAME is omitted, a list of available channels/private chats is displayed.

Called only from a script

/wait TIME

Causes a script to pause for TIME seconds. This command is *only* available for use in scripts.

Macro examples

Greeter macro

Command **/macro hi 1 /msg $WHERE Welcome to $WHERE, $1!**

Description Creates a macro to welcome someone to whatever channel you're in. Pass the nickname of whoever you want to welcome to the channel as the first argument, and it will send a public message to channel welcoming them. For example, if you were in a channel named **#erk**, and you wanted to welcome a new user named Alice to the channel, you'd execute:

/hi Alice

Which would send the following public message to **#erk**: "Welcome to #erk, Alice!"

Trout macro

Command **/macro trout 1 /me slaps $1 with a trout.**

Description Creates a macro to send the old classic message. Pass the nickname of the user you'd like to slap, and it will send a CTCP action message to the current channel or private chat. For example, to slap a user named Bob:

/trout Bob

Which would send the following CTCP action message to the chat: "[your nickname] slaps Bob with a trout.

Special attention trout macro

Command **/macro trouts * /me slaps $1 with a trout, and $+ too**

Description Creates a macro to send a new classic message. Pass the nickname of the main target, and a short list of other targets to the macro to send the new CTCP action message. For example, to slap Bob, Alice, and Dave:

/trouts Bob Alice and Dave

Which would send the following CTCP action message to the chat: "[your nickname] slaps Bob with a trout, and Alice and Dave, too"