Amiga AmIRC

A versatile IRC client for AmigaOS
Release 1.40

23.02.1997

by Laurence Walton

1 Copyright

1.1 Copyright

AmIRC has been designed, developed and written by

Oliver Wagner
Landsberge 5
45549 Sprockhvel
Germany
E-Mail: <*owagner@vapor.com*>

2 Registering AmIRC

2.1 License

AmIRC 1.40 Registered Software License Agreement
——————————————————

AmIRC 1.40 is Copyright 1996-97 by Oliver Wagner, All Rights Reserved. The producer of this software will grant the Licensee a limited, non-exclusive right to use the program AmIRC 1.40 and it's associated files on a single machine. The producer will hand out a special identification file called the key file to the Licensee. This key file will enable the Licensee to fully use the AmIRC 1.40 package according to the terms of this license. The key file is stricly for personal use by the Licensee only. Any disclosure of they key file will invalidate the license and the key file itself. The Licensee will be held liable of any damage arising out of the disclosure of the key file. The product is provided as is without warranty of any kind, either express or implied, statutory or otherwise, including without limitation any implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to use, results and performance of the product is assumed by you and should the product prove to be defective, you assume the entire cost of all necessary servicing, repair or other remediation. In no event shall the producer of this product or it's resellers be liable for any property damage, personal injury, loss of use or other indirect, incidental or consequential damages, including without limitation, any damages for lost profits, business interruption or data which may be lost or rendered inaccurate, even if we have been advised of the possibility of such damages. This agreement shall exclusively be governed by the laws of the Federal Republic of Germany.

2.2 Registration

You may use AmIRC freely for a period of up to 30 days to evaluate the package. After those 30 days, you must either register AmIRC or stop using it completely.

To register AmIRC, see the included registration utility and assorted documentation. It has all the necessary info and prices.

3 What is IRC, by the way?

3.1 Introduction

IRC stands for Internet Relay Chat, written by Jarkko Oikarinen (jto@tolsun.oulu.fi). In a nutshell, IRC consists of a network of computers (servers) that are connected to each other via the internet or a LAN, and allow users with proper client software to connect and chat with each other. Within this network, virtual space is divided up into channels, usually with a topic of conversation, where users can talk in groups.

4 Terms You Might Encounter

4.1 IRC Terms

`Action` *CTCP* Action command, executed by a /me at the beginning of a message.

`ARCnet` Amiga Relay Chat Network. Served by several Amigas running CLChatServer, forming a small network used by those interested in the Amiga Family of Computers.

`Ban` A *channel mode* set in order to prevent a *user* or group of users from joining a *channel*.

`Bot` Short for Robot, a Bot is a *script* that acts like a user, and can execute a command upon a certain occurance.

`Client` Software that is written for the purpose of connecting to a *server*, in this case, an *IRC server*.

`Channel` A group of users on *IRC*. A channel will have a name (i/e, #AmIRC), and may or may not have a *topic* of conversation.

`Channel Mode`
: The properties of a *channel* that allow or disallow certain activities. Channel Modes can only be changed or set by *Channel* Operators.

  `Invite Only`
  : If active, This mode allows users to join if they have been invited by someone in the channel.

  `Private` If active, the *channel* is invisible in the global channel list.

  `Limit` If active, Sets a limit on the number of *users* allowed to join a *channel*.

  `Moderated`
  : If active, only *Channel Operators* and *users* with *voice* may type input to the *channel*.

  `No Messaging`
  : If active, prevents *users* from sending a message to the *channel* until they have joined it.

  `Secret` If active, the *channel* is invisible until it is joined.

  `Topic Protection`
  : If active, only *Channel Operators* may set or change the *topic* of a *channel*.

`Channel Operator (aka, Op)`
: A *user* who has control over a *channel*. A user who creates a channel is given Op status, and in turn can give it to anyone else in the channel.

`Chat Network`
: A network of computers that are interconnected so that anyone on any server in the network may chat with anyone else on any computer in that same network.

  `some major chat networks:`

  `ARCnet` Amiga Relay Chat Network. Note that this is not a real IRC network and has its own client, but IRC clients can connect here, with some limitations.

  `EFnet` Eris Free Network. The largest IRC network.

- ‘DAL-Net’ A different IRC network with some specialized functons that do not appear in typical IRC networks.
- ‘Undernet’ An alternative IRC network
- ‘Innernet’ Another alternative network.

‘Command’ A message to an IRC server or to a local or remote client. Commands generally begin with a /. See Chapter 13 [Commands], page 29.

‘CTCP’ Client To Client Protocol, A standard for sending commands from one *client* to another, usually for the purpose of getting information from the remote client. See Chapter 11 [CTCP], page 25.

‘DCC’ Direct Client Connection - a standard for creating a direct link between two clients, for the purpose of sending or receiving files, or to carry a more secure conversation. See Chapter 12 [DCC], page 27.

‘Efnet’ (Eris Free), the largest and "main" *IRC network*, with an average of about 5000 *users* on any given day.

‘Identd’ Identd Is software that can be run on the client, allowing an IRC server to corroberate the identity of a client. AmIRC has an Identd built in, and will automatically disable it if an external Identd is running.

‘IRC’ Internet Relay Chat - Any *Network* of Computers that run IRC *servers* for the purpose of allowing *users* to connect to it using *client* software made for IRC.

‘Kick’ The occurance of a *user* being removed from a joined *channel* by a *channel operator*.

‘MOTD’ Message Of The Day, A message that may or may not be updated daily, that is meant to be read when connecting to an *IRC server*.

‘Netsplit’
An event that occurs when a *server* goes down that other servers depend on for network packets. Any servers that are dependant on the crashed server will split from the net, that is, they will still be up and running, with active *clients*, but any clients on the other side of the crashed unit will appear to have all logged off at once.

‘Nick’ Short for Nickname, this is an alias that you may pick for yourself.

‘Op’ See *Channel Operator*.

‘Server’ A computer that is Running IRC server software, allowing it to be connected to by *IRC clients* or other servers.

‘Server Operator’
A person in control of an *IRC server*.

‘TLA (Three Letter abreiviations)’
commonly used acronyms, not entirely restricted to 3 letters.

- ‘AFAIK’ As Far As I Know
- ‘AFK’ Away From Keyboard
- ‘BRB’ (I’ll) Be Right Back
- ‘BBL’ (I’ll) Be Back Later
- ‘IMHO’ In My Humble|Honest Opinion
- ‘LOL’ Laughs Out Loud

- '`MOTD`' Message Of The Day
- '`ROTFL`' Rolls On The Floor Laughing
- '`RTFM`' Read The F***ing Manual

'`Topic`' A heading for a *channel* that may or may not have anything to do with the conversation at hand.

'`Undernet`'
The second largest *IRC* network, considered an 'Alternative' to *EFnet*.

'`User`' a person using an *IRC Client* to connect to a *server* to chat.

'`User Mode`'
Settings that affect a users properties.

- '`Invisible`'
  If this mode is set, a *user* will be invisible in the global user list. The user will still be visible in the user list of any joined channel, however.
- '`Wallops`' When set, this mode will allow receiving of messages from the *server operator* to all operators.

'`Userhost`'
This is the string that defines a *user*. It consists of the user's chosen *nick*, login name and domain name. For example:

```
Wupper!jwupper@monorail.denet234.de
| | | | |_ Domain
| | | |________ Network
| | |________________ Machine
| |________________________ Login
|________________________________ Nick
```

'`Voice`' Status a normal (not an operator) user must have to type into a *moderated* channel.

5 Things that make AmIRC special!

5.1 Features

- runs with both AmiTCP and AS225, and now Miami, without additional support libraries.
- built-in high speed DCC drivers for SEND and MOVE. Highly optimized with asynchronous file I/O and threaded network handling. Supports Drag&Drop sending of files.
- transparent implementation of DCC CHAT. All /MSGs are automatically routed via DCC CHAT if applicable, to save runtime and reduce IRC load. Option to automatically accept DCC CHAT connections.
- ultra fast text output and scrolling. Supports Cut&Paste for the list text. Supports proportional fonts.
- full support of the "Undernet" IRC extensions, both reply codes and additional IRC commands.
- intuitive channel display with userlist and channel mode overview. Channel modes can easily be changed via gadgets. Configurable user action list to perform whatever action you like on users in channel by the press of a mousebutton.
- supports QUERY windows for private chat.
- supports "smart" banning
- versatile AREXX port including server message parsing ("event trapping") and commandline/script interaction.
- easy command line aliasing.
- builtin IdentD (automatically disabled if another IdentD is already installed).
- supports nick notification, highlighting, ignoring, and kickban.
- versatile event handling allows you to bind sound effects, auto-deiconify or rexx scripts to certain events.
- server phonebook.
- last nick history. Stores the nicknames of the last nicks you've send MSGs to, received MSGs from or addressed publically with "nick:". Reference can be used in all commands which require nick names, e.g. /MSG, /OP, /DEOP. Automatic nick expansion on public messages with ":".
- TAB messaging to automatically cycle the list of last used nicknames with a /MSG prefix.
- auto rejoin channel on KICK and auto reconnect server facility.
- auto away after idle timeout. Auto unaway after sending public and/or private msgs.
- auto update download. Check for and download new releases with a simple click.
- URL grabber for interfacing with a web browser..
- Support for DALnet extensions.

6 What is needed to run AmIRC?

6.1 Hardware

AmIRC was written for the Amiga family of computers, and has been tested on 1200, 2000, 3000, & 4000 computers in various configurations.

6.2 Software

AmIRC requires Chapter 25 [About MUI], page 69 V3.6 or later in order to run.

AmIRC requires KickStart & WorkBench 2.x or later, with setpatch 40.16 installed.

AmIRC020 requires a 68020 or higher CPU, OS3.0 or newer, and MUI 3.6 or newer. If you do not have anyone of these, you must use the 68000 version of AmIRC.

If you plan to run a TCP/IP stack and AmIRC among other Internet applications you should have at least 2 megabytes of RAM.

Finally, You need to have a tcp/ip protocol stack running, such as Miami, AmiTCP/IP or Amiga Technology's as225r2 tcp/ip protocol stack, With a slip, cslip, or ppp connection to the internet or an ethernet LAN.

Alternatively, AmIRC will work with Multilink and a shell internet account. NOTE: As Mlink does not automatically set the HOSTNAME variable, this must be done when starting mlink. Env:hostname must contain the full domain name for your amiga, i/e 'calci.amiganet.pgh.net'.

7 Getting AmIRC to run

7.1 Requirements

AmIRC has its own installer script and requires that you have the Amiga installer utility (version 1.24 or later) located in your sys:c directory. To install, just click on the install icon, and answer the questions in the installer's requestors.

If you cannot or prefer not to use the installer utility, a manual install is very simple, just move the contents of the archive to a directory of your choice and it's ready to use. If you have met the Chapter 6 [Requirements], page 13 for the optimized version of AmIRC, delete the AmIRC file and rename "AmIRC020" as "AmIRC".

8 Connecting to a server

8.1 Setup

Once you have Chapter 7 [Installation], page 15 AmIRC, run the AmIRC program from WB by clicking on its icon. After a few moments you will see the server connect requester appear. First type in a nickname in the first text gadget at the top of the requester. If you intend to use a nick that may be common, try typing in some variations in the next three boxes. this way, if your nick is in use by another person, the client will retry with the next nick.

Next, enter your real name and your email address or www home page URL in the next text gadget. It is probably a good idea to be truthful here. :)

Then, enter a username that will show as username@machine.network.org. username should be the same as the login ID for your service provider.

Now you are ready to connect. Just click on a server from the supplied list or click new server entry and type the address and port number of your favorite server and then click connect. Optionally, you can choose one or a few channels to autojoin, just separate them with a comma. If you are already connected, and wish to connect to a second server, click connect thread to open a second instance of AmIRC. Use the Copy Server Entry gadget to make a duplicate of the currently selected server entry. The Delete Server Entry gadget will remove a server entry that you no longer wish to use.

9 How Does AmIRC Work?

9.1 Using the Interface

The AmIRC user interface is laid out in a logical, intuitive manner, making it easy to use.

'`Text Listview`'
: The main feature of the interface is the text listview, where incoming text is displayed.

'`Text Input Gadget`'
: Directly below this is the text input gadget, labeled with the nickname that you have chosen. Any text entered here will be sent out to the IRC server.

'`Listview Scroll Gadget`'
: To the right of the listview is a scroll gadget, which enables you to scroll backward through the text in the listview.

'`Channel Text String`'
: At the top left of the listview is a text string that reflects the channel that you have joined.

'`Topic Input/Display Gadget`'
: To the right of the channel text string, is the Topic Input/Display gadget, where the topic of a joined channel will be displayed. If you have Channel Operator (Ops) Status, you may change the topic of the channel by entering the topic in this gadget, and hitting the 'enter' key.

'`Operator Buttons`'
: To the right of the topic input gadget are the channel mode buttons, which display the current modes for the joined channel. Users with Channel Operator (Ops) status may also use these buttons to change the current channel modes.

  '`T - Topic Protection`'
  : Toggles Topic Protecton on/off. If on, the topic can be changed by Channel Operators (Ops) only.

  '`N - No Messaging`'
  : Toggles Messaging on/off. If active, a user can only write to the channel if they have joined it.

  '`S - Secret`'
  : Toggles Secret mode on/off. If active, the channel is invisible until it is joined.

  '`I - Invite Only`'
  : Toggles Invite Only mode on/off. If active, users can only join after they have been invited.

  '`P - Private`'
  : Toggles Private mode on/off. If active, the channel will be invisible in the global channel list.

  '`M - Moderated`'
  : Toggles Moderated mode on/off. If active, only channel operators (Ops) and users with 'voice' can type to the channel.

`L - Limit`
: Sets a limited number of users allowed on the channel

`K - Keyword`
: Toggles Keyword mode on/off. If active, Users can only join the channel if they give the correct keyword.

`B - Ban`
: Sets/Displays the current Bans active on the channel

`User Listview`
: Below the channel mode buttons is the User Listview, which shows a current list of users on the joined channel. Double clicking on a users name will perform a /whois on that user. Clicking once on a users name will set that user as the default to perform other operations on, such as an operation from the configurable user buttons.

`User Buttons`
: Directly below the User Listview are the user buttons. These buttons are configurable from the Chapter 20 [GUI], page 53 window, accessable by selecting 'setup...' from the AmIRC Section 10.1.4 [Settings], page 23 menu.

`Special Functions:`
: You can Drag-Hilight text in the listview in order to paste it elsewhere. you will need some type of snap utility such as newedit or powersnap in order to past the text. The text is stored in the clipboard.

The `TAB` key will cycle through the names of people that have previously /msg'ed you, and insert it into the text input gadget. Simply enter your message after the "/msg <user> " in order to /msg them back.

Entering ":" first in the text input gadget will automatically publicly address the last person you typed to using the ":" after their nick.

10 Pull-Down Menus

10.1 Reference

10.1.1 AmIRC Menu

'`Select Server...`'
: Opens a window from which you can select various options for connecting to an IRC server. See Chapter 8 [Connecting], page 17.

'`Logging Enabled?`'
: Toggles logging on/off.

'`Iconify`' Reduces the AmIRC window to an icon on the Workbench screen.

'`Help...`' Opens Amirc.guide for reference while using AmIRC.

'`About AmIRC...`'
: Opens a window with information about AmIRC and its Developer. Here you may connect to the AmIRC home page if amosaic is running concurrently, and you may also download the latest version of AmIRC without having to trouble with running a www or ftp client.

'`About MUI...`'
: This will open a small window that displays information about the current version of MUI that is installed on your system, also, this window provides the option to register or update your MUI software.

'`Quit`' Will close the current instance of AmIRC, leaving other instances open.

10.1.2 Windows Menu

'`Channel Windows`'
: If more than one AmIRC window is opened, You may select one here to pop it to front.

'`Dialog Windows`'
: If any dialog windows are open, you may select one here and bring it to front.

'`Channel List Window...`'
: This item will produce a requester where you may list channels by min users or pattern matching, or both. Once listed, you may list users on a channel or join it by double-clicking the channel name or selecting 'join channel' gadget. Please note that this function can cause a high load of traffic and may take awhile on slow (14.4 or less) connections.

'`Notify Window...`'
: Selecting this item will produce a requester where you may enter nicks of users that you wish to be notified of their entering/leaving IRC. The 'check period' is how often AmIRC will check for the specified users. Add a nick to the list by clicking in the input text gadget, and typing in the nick. delete a nick by clicking on it's entry and then click on 'delete nick' gadget.

'DCC-Send Window...'
: Brings up a requestor in which you can specify a user and a file to send to them. This requester will also be produced automatically upon hitting the 'DCC send' userbutton on the interface. 'Send Path' is the path and filespec that will be sent to the user. Enter the name of the user in the text input gadget at the bottom of the requestor. The cycle gadget nest to the nick text gadget selects either 'DCC Send' or 'DCC Move' protocols. See Chapter 12 [DCC], page 27 for details about these protocols. 'Remove Job' gadget will remove a complete send entry or if not complete, one click will abort the send and the second click will remove the entry.

'DCC-Receive Window...'
: Selecting this item will produce the 'Receive DCC List' requester. This is very similar to the DCC Send Requester. Note that this requester will appear automatically on receiving a DCC request. Enter a path for incoming files to be put in the 'Receive Path' text gadget. After a valid path is entered, click on an entry to receive, then click the 'Get' gadget at the bottom. If you are trying to receive after a failed send, you may use the 'Resume' gadget instead of get. this will attempt to pick up the transfer where it left off. I you wish to rename the incoming file for whatever reason, you may use the 'Rename' gadget to give the file a new name. 'Remove Job' gadget will abort an incoming file and remove a completed or aborted file. See Chapter 12 [DCC], page 27.

'DCC-Chat Window...'
: Selecting this item will open the 'DCC Chat List' requestor. Do this if you receive a DCC chat request from another user.. Here you may click on the client sending the chat request and click the 'Open' gadget to engage a DCC chat. When finished, you can click "Close' to close the connection. See Chapter 12 [DCC], page 27.

'Raw Log Window...'
: This will open the Raw Log Window, in which you can view the raw IRC messages as they appear before AmIRC formats them for Easy reading in the AmIRC listview. Here you may also clear the log, and select when the log is active: 'Never', 'Only when raw window open', and 'Always' The log Listview shows the USERHOST, Type of message, Channel Name, and message, in that order.

'URL Grabber Window...'
: A New feature of AmIRC allows it to recognise WWW URLs and snap them to a window for use with Amosaic Web Browser. Double clicking an entry in the URL grabber Window will automatically send the URL to the Clipboard..

'Private Messages Window...'
: This item will open the Private Messages Window. This window holds a history of private messages sent to you, where you can later reference or review them.

'Info Messages Window'
: this item will open a window that displays server messages, wallops messages, and ctcp requests. If the "Shangahi" gadgets are activated for these messages, they will be prevented from showing in the main channel window, even if the info window is closed.

'Flush Buffer'
: This item will clear the history of the main AmIRC listview.

'Save Buffer...'
: This item allows you to select a filename to save the history buffer to disk.

10.1.3 User Modes Menu

'`Invisible?`'
: Toggles your client between invisible or not invisible. See Chapter 4 [IRC Terms], page 7.

'`Receive Wallops?`'
: Toggles this mode on/off. See Chapter 4 [IRC Terms], page 7.

'`Receive Server Notices?`'
: Toggles this mode on/off.

'`Marked AWAY?`'
: Toggles AWAY on/off.

'`Auto rejoin on KICK?`'
: If set, will automatically attempt to reconnect you to a channel you have been kicked from. Note that if a ban was set with the kick, you will not be able to rejoin.

'`Auto reconnect to server?`'
: If set, this will automatically attempt to reconnect you to the server if you are disconnected for any reason.

'`Auto open Dialog Windows?`'
: If set, will automatically open chat window upon receipt of a /msg or dcc chat message.

'`Auto accept dcc chat requests?`'
: If set, this will automatically accept a dcc chat request. It will not open a chat window for you, however. Do this by opening the DCC chat window in the Windows menu.

'`Do BEEPS?`'
: If set, AmIRC will send audio beeps (or samples, if configured in the Chapter 19 [Events], page 51 page of the settings requester) for certain events.

'`Play CTCP SOUND requests?`'
: If checked, AmIRC will acknowledge and play /ctcp sound requests, if this is configured properly.

10.1.4 Settings Menu

'`Setup...`'
: Calls the Chapter 16 [AmIRC Setup], page 45, allowing you to change the many settings of AmIRC to suit your needs.

'`Lists...`'
: Calls up the Chapter 22 [Lists], page 57, allowing you to configure AmIRC's Alias, Ignore, and Hilite lists.

'`Function Keys...`'
: Calls the Chapter 23 [Function Keys], page 59, allowing you to configure the function key macros.

'`Dialog Window Buttons?`'
: If set, the AmIRC dialog window will contain 4 gadgets in the upper right corner. These are labeled: 'Whois', 'Ping', 'DCC Chat', & 'DCC Send'. These gadgets will perform their respective functions on the user that the dialog window is opened to.

'`Load Settings...`'
: Opens a requestor allowing you to recall previously saved non-default AmIRC Configurations.

'`Load settings from default`'
: Reloads the last saved default configuration

'`Save Settings Saves the current configuration under the last name set`'
: in the 'Save settings as...' menu item. If the settings have not previously been named, they are saved as default.

'`Save settings as...`'
: Opens a requester allowing you to choose a filename to save the current AmIRC configuration as other than default.

'`Save as default`'
: Saves the current AmIRC configuration as the default startup configuration.

'`Create icons?`'
: If Checked, will create icons for saved configuration files, allowing them to be seen on the Workbench.

'`Save Settings On Exit`'
: if checked, this item will cause AmIRC to save the current settings as default upon exiting AmIRC.

'`MUI Settings...`'
: Opens the MUI Prefs program for altering the MUI Configurable AmIRC configuration.

10.1.5 Rexx Menu

This menu will Display any ARexx scripts that reside in AmIRC/Rexx/ directory. AmIRC ARexx scripts should be named with the .amirx extension. From this menu you may execute any existing AmIRC ARexx script.

11 Sending commands to other people

11.1 Introduction

11.1.1 CTCP or *Client To Client Protocol*

CTCP is a standard for sending commands to other clients on IRC. Basically, CTCP messages are simply private messages that include a code to alert the receiving client (not user) to respond to the command.

CTCP commands are typicaly used to get informaton from the receiving client, such as /CTCP time, or /CTCP version. Even Chapter 12 [DCC], page 27 sends a request via CTCP to the receiving client, asking to initiate a direct connection.

11.1.2 CTCP Commands

‘

`ME (CTCP Action)`’

Template: /ME <*action*>

This will mark the message as an action and replace the /ME with your nick, for example, ’/me hates ice cream’ would appear as:

‘`[Action]`’ |Fred hates ice cream

‘

`Version`’

Template: /CTCP <*user*> *Version*

Version will ask the client to return information about itsself, which will be in the form of a private message.

‘

`Time`’

Template: /CTCP <*user*> *Time*

Time will return the time of day as it is set on the user’s machine.

‘

`Ping`’

Template: /ping <*user*>

Ping will send a packet request to the receiving client, which will in turn send back the packet with a timestamp, where the inititating client will compare the time difference and show the result in the listview.

‘

`Sound`’

Template: /CTCP <user> Sound <soundfile>

Sound will attempt to locate and play a soundfile in the directory the receiving user has configured as the "sound" directory in AmIRC. Some IRC clients for other platforms support this function as well. note that the <soundfile> must exist in the receiving users directory or nothing will happen. Generally, a set of sounds are agreed upon in advance for a particular group of users.

12 Direct client connections

12.1 What is DCC?

DCC or Direct Client Connection is a protocol that IRC clients can use to communicate directly with each other, bypassing the IRC server. The reason for this is to form a faster, more stable and direct connection thru which users can transfer files, or send private messages.

Note that AmIRC has its own set of Highly Optimized DCC protocols built-in, and does not require external DCC utilities.

12.2 DCC File Transfers

The most common use of DCC is for transferring files between two clients. The initiating client will send a message through the IRC server, telling the receiver that a connecton is requested. At this time, the receiver will either accept or reject the offer for the connection. If the receiver accepts the connection, the initiating client will proceed to send the information directly to the receiver. If the connecton is refused, the initiating client will simply timeout the attempt to connect.

12.2.1 DCC Send

Template: /DCC send <user> <filespec>

DCC send will attempt to open a connecton with the specified user's client, and if successful, will proceed to send the specified file.

12.2.2 DCC Move

Template: /DCC move <user> <filespec>

DCC Move is identical to DCC Send, with the exception that DCC move is capable of resuming an interrupted or failed transfer. DCC Move will look into the receiving directory for existing files, and upon finding a partial file which is the same as the file to be transferred, will initiate a /DCC Resume transfer. '`With AmIRC, This process is completely transparent`'. If no partial or whole file exists, then the file will be completely transferred.

12.2.3 DCC Get

Template: /DCC get <user sending file>

> DCC get is a transparent operation with AmIRC, and is only mentioned in this document for the sake of being thorough. AmIRC will notify you of an incoming DCC request, and allow you the option to accept or reject the connecton with a simple click.

12.3 DCC Chat

Another use of the DCC standard is DCC chat, which, like DCC Send, will send a CTCP request to the receiving client to set up a direct link. In this case, however, the purpose of the link is for users to send messages to each other. Because the mesages are sent through the direct link, and do not pass through the IRC server, this method of sending messages is much more private than using the IRC's privmsg (/msg).
With AmiRC, it must be noted that a DCC Chat request will not open a dialog window for you, instead, when DCC chat is initiated and accepted, all privmsg's (/msg) to the receiving user will be sent through the DCC link. If you wish to have a dialog box for chatting, the best way to go about this is to highlight the users's name in the listview by clicking on it. then click on the 'Query' gadget below the listview. this will open a window that has several options within it. From here, you can 'ping' or 'whois' the other user, or you may inititate a 'DCC Send' or 'DCC chat'. Starting a 'DCC Chat' from this window will allow you to carry on a dialog with the remote user through the window.

13 Commands

13.1 Overview

AmIRC has several builtin commands. Some of these commands are processed locally, whereas others are sent to the IRC server.

Commands generally start with a `/` to differentiate them from normal text. If you want to use the `/` character at the start of normal text, use `//`instead.

Many commands require a channel name as a parameter. AmIRC will automatically fill in the channel name of the current window if you don't specify a channel name on your own.

For commands which require a nick name (e.g. '`/MSG`'or '`/OP`', there are three special shortcut arguments:

- `.` The period is replaced with the nickname of the last person you have send a private message to.
- `,` The comma is replaced with the nickname of the last person who send a message to you. Be careful with this, since you might easily address the wrong person if someone sends you a message while you are still typing this command!
- `:` The colon is replaced with the nickname of the last person which you haved a addressed with a `nick:` prefix inside a channel.

You may easily carry a dialog with a particular user by first clicking on their username in the user list, then anytime you hit the tab key, the input string will be set up with '/msg <user> ', after which you may type in the message and hit return. This will work after initiating a Chapter 12 [DCC], page 27 Chat request as well. Please note that if you have configured the user list as a separate window, you must reactivate the message window after clicking on a username.

13.2 Reference

13.2.1 AWAY

Usage: `/AWAY [message]`

When used with the optional <message> argument, /AWAY will mark you as being away and send the message to any client sending a Section 13.2.12 [MSG], page 34 to you. otherwise, it will remove a previously set away message.

EXAMPLES:

'`/AWAY eating dinner`'
will send the message '`<nick>` is away: eating dinner', where nick is the nickname of the user that is away.

'`/AWAY`' will remove a previously set Away Message.

13.2.2 INVITE

Usage: `/INVITE <nickname> <channel>`

Invites a user to the channel given in the argument, regardless of weather the channel exists. If the named channel is Section 13.2.10 [MODE], page 33, the user performing the command must be a channel operator for that channel.

EXAMPLES:

'`/INVITE freddie #AmIRC`'
Sends a message to 'freddie', inviting him to join #AmIRC.

13.2.3 ISON

Usage: `/ISON <nickname>{<space><nickname>}`

Quickly returns the users nick if it is found, does nothing if it is not.

EXAMPLES:

'`/ISON Kenneth`'
Returns: "`<ISON>`'Kenneth', if nick 'Kenneth' is found on the server.

13.2.4 JOIN

Usage: `/JOIN channel[,channel[,...]] [key[,key[,...]]]`
Usage: `/CHANNEL channel[,channel[,...]] [key[,key[,...]]]`

Join a channel on the IRC. If the channel doesn't exists, it is automatically created.

There are two kinds of channels: Channel names starting with an #are global channels and available on all IRC servers which are connected to the net you are currently on. Channels which begin with &are local to the chat server you are connected to.

If a channel requires a special keyword to join, you must submit this after the channel name.

AmIRC will look for a channel window which isn't currently bound to a channel. If it doesn't find one, a new window will be created and bound to this channel.

EXAMPLES:

'`/JOIN #Amigager`'
: Join channel *#AmigaGER*

'`/JOIN #Amigager,#AmIRC`'
: Join channels *#AmigaGER* and *#AmIRC*

'`/JOIN #AmigaGER deutsch`'
: Join channel *#AmigaGER* with keyword *deutsch*.

RELATED TOPICS:

See Section 13.2.7 [LEAVE], page 31.

13.2.5 KICK

Usage: `/KICK <channel> <user> [<comment>]`

Forcibly removes a user from the named channel. the user sending the /KICK command must be a channel operator of the named channel. If the optional `<comment>` is given, it will be shown as the reason for the kick.

EXAMPLES:

'`/KICK #AmIRC Freddie Take it outside!`'
: forces 'Freddie' from channel #AmIRC with the reason 'take it outside'.

13.2.6 LASTLOG

Usage `/LASTLOG <text>`searches back through AmIRC's buffer and displays all occurances of the specified text.

EXAMPLES:

'`/LASTLOG amirc`'
: searches and displays all lines from the buffer containing the text "amirc"

13.2.7 LEAVE

Usage: `/LEAVE [channel[,channel[,...]]`

Leaves a channel. If you don't give a channel name as a parameter, the channel currently bound to the window in which in the command is entered will be left.

EXAMPLES:

'/LEAVE' Leave the channel currently bound to this window.

'/LEAVE #AmigaGER'
Leave channel *#AmigaGER*.

RELATED TOPICS:

See Section 13.2.4 [JOIN], page 30.

13.2.8 LIST

Usage: `/LIST [<channel>{,<channel>} [<server>]]`

Generates a list of current channels and topics. If the `<channel>` argument is given, LIST displays that channel's status and topic. Private channels show as 'Private' and secret channels are not listed, unless the client generating the list is a member of those channels.

EXAMPLES:

'`/LIST`' Shows a list of all current, non-secret channels.

'`/LIST #AmIRC,#amigaGER`'
List channels #AmIRC and #amigaGER

13.2.9 Logging Options

Usage: `/LOG {[on|off|<Filespec>]}`

/LOG allows you to start/stop logging, as well as setting the log filename and displaying status of logging.

EXAMPLES:

'`/LOG`' Displays current status of logging

'`/LOG on`' Begins logging to the logfile

'`/LOG <filespec>`'
Sets the path and name of the logfile.

13.2.10 Channel MODE

/MODE allows both users and channels to have their modes changed.

Usage: `/MODE <channel> {[+|-]|o|p|s|i|t|n|b|v} [<limit>] [<user>] [<ban mask>]`

Used by Channel Operators to change the modes of 'their' channel. the modes available are:

o - give/take channel operator privileges;
p - private channel flag;
s - secret channel flag;
i - invite-only channel flag;
t - topic settable by channel operator only flag;
n - no messages to channel from clients on the outside;
m - moderated channel;

l - set the user limit to channel;
b - set a ban mask to keep users out;
v - give/take the ability to speak on a moderated channel;
k - set a channel key (password).

EXAMPLES:

'`/MODE #AmIRC +tm`'
: Sets topic protection and makes channel moderated for #AmIRC.

'`/MODE #AmIRC -o Sarek`'
: Removes channel operator status from user 'Sarek' on channel #AmIRC.

'`/MODE #AmIRC +l 50`'
: Sets limit of 50 users on #AmIRC.

'`/MODE #AmIRC -l`'
: Removes the limit form #AmIRC

13.2.11 User MODE

Usage: `<nickname> {[+|-]|i|w|s|o}`

These affect how a client is seen by others or what extra messages a client will receive. User Modes will only take effect if the Nick of the client sending the command, and the nick of the `<nickname>` argument are the same.

Available User Modes:

i - marks a users as invisible;
s - marks a user for receipt of server notices;
w - user receives wallops;
o - operator flag.

Note that the '`+o`' user mode will not work, however, '`-o`' will.

EXAMPLES:

'`/MODE freddie +i`'
: makes user 'freddie' invisible in the global user list.

'`/MODE freddie -o`'
: User 'freddie' removes channel operator status from himself.

13.2.12 MSG (PRIVMSG)

Usage: `/MSG <receiver>{,<receiver>} <text to be sent>`

Sends a Private Message to one or more users. Note that this is different from a Chapter 12 [DCC], page 27, in that a /MSG (PRIVMSG) is sent thru the IRC server to the recipient. Note that the 'TAB' key will cycle back and forth through the users who have /msg'ed you. This will appear in the input string gadget, and you may simply add your message after it.

EXAMPLES:

'`/MSG Kenneth Heya Dude! How's Florida?`'
Send the message: "Heya Dude! How's Florida?" to user 'Kenneth'.

'`/MSG Kenneth,Aerosoul I am in Pittsburgh`'
Sends the message "I am in Pittsburgh" to Kenneth and Aerosoul.

13.2.13 Names

Usage `/NAMES [<channel>{,<channel>}]`

Gives a list of nicknames on any channel visible to the user(not secret or private, or one actually joined by the user performing the command).

EXAMPLES:

'`/NAMES #AmIRC,#amigaGER`'
Will list all users on channels #AmIRC and #amigaGER, provided the channels are visible to the user.

'`/NAMES`' Will return a list of all visible channels and users (Strongly Discouraged!)

13.2.14 NICK

Usage: `/NICK <nickname>`

NICK is used to to give yourself or change your current nickname. If the chosen nick is already in use, you wil be notified of this and prompted for a new nick.

EXAMPLES:

'`/NICK Phalanx`'
Makes/changes your nickname to 'Phalanx'

13.2.15 NOTICE

Usage: `/NOTICE <nickname> <text>`

Smiilar to /MSG, /NOTICE is usually used by Bots or clients being controlled by scripts. /NOTICE messages must never be responded to by an automatic reply. See Section 13.2.12 [MSG], page 34 for syntax examples.

13.2.16 QUIT

Usage: `[<Quit message>]`

Used to End an IRC session. The optional `<`quit message`>` is the text that will be displayed upon quitting.

EXAMPLES:

'`/QUIT :Gone to have lunch`'

13.2.17 RAW

Usage: `/RAW text`

Sends text directly to the IRC server without any further processing. **Warning:** Don't even think about using this command without interior knowledge of the IRC protocol, or you might run into severe trouble. See *RFC 1459* for a description of the IRC protocol.

EXAMPLES:

'`/RAW JOIN #IRC-Heros`'
Send the string *JOIN #IRC-Heros*directly to the IRC server.

RELATED TOPICS:

RFC 1459.

13.2.18 TIME

Usage: `/TIME [<server>]`

Returns the time of day from the server. If the optional <server> argument is given, then that servers local time will be returned.

EXAMPLES:

'`/TIME irc.pitt.edu`'
: Returns the local time from server 'irc.pitt.edu'

13.2.19 TOPIC

Usage: `/TOPIC <channel> [<topic>]`

The topic command alone will display the channels topic. if the optional <topic> argument is given, the topic will be changed to the argument given, provided that the topic is not protected, or the user issuing the command is an operator on the channel. See Section 13.2.10 [MODE], page 33.

EXAMPLES:

'`/TOPIC #AmIRC`'
: Will display the current topic on #AmIRC.

'`/TOPIC #AmIRC Amigas Rule!`'
: Will replace the current topic of #AmIRC with the topic 'Amigas Rule'

13.2.20 USERS

Usage: `/USERS [<server>]`

Returns a list of users on the server, or, optionally, on the specified server. /USERS may be disabled on most servers for security reasons.

EXAMPLES:

'`/USERS`' Returns a list of users on the current server

'`/USERS irc.colorado.edu`'
: Returns a list of users on server 'irc.colorado.edu'.

13.2.21 USERHOST

Usage: `/USERHOST <nickname>{<space><nickname>}`

Returns the login@domain for the specified nickname. Up to five nicknames may be specified, each separated by a <space>.

Examples:

`/USERHOST DuRaN`
: returns: 'DuRaN is DuRaN@calci.amiganet.pgh.net

13.2.22 VERSION

Usage `/VERSION [<server>]`

Returns the version of the server's software. If the optional <server> arguement is given, returns the version of the specified server's software.

Examples:

`/VERSION *.edu`
: Returns server version information for all servers matching '*.edu'.

13.2.23 WHO

Usage: `/WHO [<name> [<o>]]`

Returns a generated list of all users matching the criteria passed to the command, i/e. everyone at a particular site, or on a particular server. If the optional <o> argument is specified, /WHO will only return information on users that are channel operators.

Examples:

`/WHO *.cmu.edu`
: Will list all users at the cmu.edu domain.

`/WHO irc.pitt.edu o`
: will list all operators on irc.pitt.edu server

13.2.24 WHOIS

Usage: `/WHOIS [<server>] <nickmask>[,<nickmask>[,...]]`

Returns information about the specified user or users, that you are allowed to see. If the optional <server> argument is specified, then information aobut the user in relation to that specified server will be given, i/e idle time.

Examples:

'/WHOIS Phalanx'
: Returns available information about user 'Phalanx'.

'/WHOIS irc.bu.edu Phalanx'
: Asks server 'irc.bu.edu' for information on user 'Phalanx

13.2.25 WHOWAS

Usage `/WHOWAS <nickname> [<count> [<server>]]`

returns information on a nick that no longer exists, due to the user leaving the server or changing their nick. The optional <count> argument specifies how many previous instances of the nick should be returned. The optional <server> argument specifies a particular server to be searched.

EXAMPLES:

'/WHOWAS Aerosoul 5'
: Checks for and returns available information on the last 5 instances of the nick 'Aerosoul'.

14 Customizing your environment

14.1 Overview

Now that you have AmIRC running, you may want to get it to operate the way you want it to, and to look pleasing as well. This is achieved thru both MUI preferences and AmIRC's Menu options.

15 Changing MUI Settings

15.1 MUI Preferences

Most of the configuration dealing with how AmIRC's GUI appears can be customized with Chapter 25 [About MUI], page 69. Check the 'MUI Settings...' entry under the 'Settings' menu. This will pop up MUI's interface, allowing you to setup AmIRC's MUI Prefs, without disturbing any of your other MUI configurations. Read the documentation for MUI to Learn more about the operation of MUI Preferences.

16 Changing AmIRC Options

16.1 AmIRC Setup

Most of the setup functions appear in the 'Setup' entry under the 'Settings' & 'Lists' menus. Here you will get a multi-requestor that will allow you to access several pages that deal with configuring AmIRC.

17 Messages Page

17.1 Misc Messages

'`CTCP Userinfo`'
: The text in this string is what will be returned to a user who sends '/CTCP userinfo' to your client.

'`KICK Reason`'
: If you are a channel operator and /kick another user from the Channel, this test will appear as the reason for the /KICK.

17.2 Quit Messages

'`Exit`' This will appear to users in the channel when you exit AmIRC.

'`New Server`'
: This text will appear to users in the channel when you select 'Select Server' in the 'AmIRC' menu, and pick a new server to connect to.

'`System Reset`'
: This is the text to appear if you reboot your amiga (Crtl- amiga-amiga) while connected to a server.

17.3 AWAY Messages

'`Default`' This is the text that will appear when you are set 'away' by selecting the 'Marked Away?' item in the 'User Modes' menu.

'`AutoAWAY`'
: text that appears when you are automatically set away due to inactivity.

'`Timeout`' This sets how much time will pass before AmIRC assumes you are away from the computer and marks you as 'AWAY'. If set to 0 (Zero), timeout will be disabled.

'`UnAWAY`' This cycle gadget allows you to choose the conditions of which AmIRC will automatically set you to 'Not AWAY'.

18 Actions Page

18.1 Actions

The 24 text gadgets in the 'Label:' column here correspond to the gadgets under the user list on AmIRC's main window. You may set these gadgets to be anything, sort of like macros. AmIRC, however, does include configurable function keys to act as macros, so these gadgets are preset for /ctcp, /dcc, and /mode operations.

Below the text gadgets is a quick reference of the placeholders that can be used in the desired operation. These actions can also be accessed by clicking the right mouse button in the title bar of the AmIRC screen after clicking on a users nick with the left mouse button.

19 Events Page

19.1 Events

Here you may define certain actions to take place upon incoming events. These actions can be as simple as a warning beep from your amiga, or a complex and powerful rexx command. The following text first explains the functions of the columns of this page, and then goes on to explain what each event is.

19.1.1 Columns

'`Event`' The particular event for the action to take place on.

'`Deiconify`'
: If AmIRC is iconified, this will uniconify and bring to front the AmIRC window on the occurence of the event.

'`Beep`' This will perform a system beep on the occurence of the event

'`Sound`' if a sound sample is preferred for the event, you may use the requestor gadget to pick a sound to play instead of the system beep. Please Note, AmIRC uses the DataTypes of OS 3.x for sounds, and you will not be able to use this feature under OS 2.x.

'`Vol`' This sets the volume for the sound sample. The valid range for the volume is 0 - 64. This is a slider gadget. The small gadget just to the left of this silder gadget can be used to test the selected soundfile.

'`Rexx Command`'
: Here you may configure a rexx script to be run on the occurence of the event.

19.1.2 Events

'`Any Message`'
: This Event occurs on any incoming public message.

'`Private Msg`'
: This event occurs when you receive an incoming private /msg or /dcc chat message

'`Notice`' This event occurs when an operator sends out a /notice

'`CTCP request`'
: This event occures when your client receives any /ctcp request (for example /ctcp time, or /ctcp version

'`DCC request`'
: This event occurs on receiving a /dcc file transfer request or chat request.

'`Join`' This event occurs when anyone /joins the channel you are currently on.

'`Kick`' This event occurs when anyone is /Kicked from the channel you are currently on.

`Mode change`
: This event occurs when the channel mode is changed for the channel you are currently on, i/e. if someone is /OPed or /DeOPed, or if the channel is made private, etc.

`ISON Notify`
: This event occurs when AmIRC 'sees' a user that is configured in your 'Notify' list.

`Connection`
: This event occurs when AmIRC makes a connection to an IRC server.

`Invite`
: This event occurs when another user invites you to join a channel that is currently set to /MODE 'Invite Only'

`Highlight`
: This even occurs upon receiving text that is configured in the highlights secton of the lists setup.

20 GUI Configuration

20.1 GUI

The GUI configuraton page allows you to control some specific aspects of AmIRC that arent covered in the MUI configuration.

20.1.1 Userlist:

The '`userlist`' cycle gadget allows you to control if the User List is visible or not visible, and if it is visible, if it is a separate window, or a part of the AmIRC window. If the '`Sort Gadget`' is check marked, the list will be sorted alphabetically, otherwise, the list will reflect the order that the users have joined the chat.

The '`Order`' cycle gadget allows you to choose how the nicks in the userlist will be listed, Alphabetically, new at top, or new at bottom.

The '`Buttons?`' Gadget will toggle on & off the gadgets in the User Action List, located below the user list. If the user list is configured as a separate window, the buttons will also appear in that window.

The '`Userlist win open?`' gadget will control if the userlist is open or closed if configured as a separate window.

20.1.2 Misc

Here you may select a different pubscreen for the query window to appear on. If no pubscreen is specified, the Query window will open on the AmIRC default screen.

20.1.3 Text Colors

In this section, you may configure the colors that certain text will appear in AmIRC. These colors are based on the MUI pen colors that you have configured in MUI. Just click and hold on the cycle gadget to see a list of the MUI Pen colors available.

'`Normal Text`'
: Text that is a normal message from a user.

'`Private Messages`'
: Text that is a private message from another user to you.

'`Mode Changes`'
: Text that shows a mode change for the channel.

'Highlighted Text'
: text that is configured in the Chapter 22 [Lists], page 57 list for AmIRC.

'Server Messages'
: Text that comes from the server, i/e when a user joins or exits the channel.

'CTCP/DCC Messages'
: Text that informs you of an incoming CTCP or DCC request.

20.1.4 Logging

Here you may enter the path & name to use for the logfile.

21 Misc Configuration

21.1 Miscellaneous Configuration Items

21.1.1 CTCP COUND handling

Here you may set the directory where AmIRC looks for files requested by incoming '/CTCP Sound' requests. Also, the volume of the sound to be played back can be set.
If the "Use external player?" gadget is checked, AmIRC will attempt to use the player specified in the "player command line" gadget to play sounds. This player applies to ctcp sound requests as well as sounds configured in the setup/events tab.

22 "Lists..." Menu Item

22.1 Lists Configuration

22.1.1 Alias Tab

Here you can create aliases, or shortcuts for complex IRC commands. Some examples have been created for you, and a reference template appears at the bottom of this page. %p is the place holder for any arguments you may wish to add when using an alias.

22.1.2 Ignore Tab

Here you can set which users if any you want to ignore, and what kind of messages from them to ignore. To add an ignore to the list, click the 'Add Ignore' gadget, then type the user's name in the text input gadget labeled 'Ignore'. Next click on the type of messages you want to ignore. Checking the 'text' box will prevent you from seeing any public messages from the user. Setting the 'privmsgs' box will prevent you from seeing /MSGs from that user to you. Likewise, clicking the 'CTCPs' box will prevent you from seeing any CTCP requests from the Ignored user. To delete an ignore, click on the entry you wish to delete, and then click on the 'delete ignore' gadget.

22.1.3 Hilite Tab

Here you may set certain text to be highlighted in order to get your attention if that text appears. You may set text to match the nick, Userhost, or any keyword enclosed in double quotes. To add a new pattern, click on the Button labeled "Add new Hilite pattern", and type your pattern into the input gadget. To delete an existing pattern, Click on the entry yo uwish to delete and then click on the button labeled "Delete Hilite pattern". The Hilite patterns will ignore a pattern it recognizes if it was input by you, but will activate on the pattern from other users.

22.1.4 Pattern Definitions:

`Nick`
: This will watch the AmIRC listview for the nick that you specify, and highlight the text for that users message. The 'Nick' entry does not require double quotes.

`Userhost`
: This will watch for messages that originate from a particular Userhost, and highlight the text for that message. Userhost entries may contain wildcards. Examples:

  `fred!fdoe@doe1.powernet.edu`
  : This will highlight only messages from fred at that particular address.

  `fred!*@*`
  : This will highlight messages from anyone named fred, regardless of address or login name.

'Fred!*@*.powernet.edu'
: This will highlight messages from anyone named fred on any machine on powernet.edu

'*!*@doe1.powernet.edu'
: This will highlight any messages from anyone on the doe1 machine at powernet.edu

'*!*@*'
: This will highlight any messages from any user.

'"Keyword"'
: This will highlight any message that contains the text enclosed in double quotes.

23 Function Key Definitions

23.1 Function Keys

Here you may define the Amiga's ten functon keys to insert text into the input string gadget of AmIRC. By using the keys normally, or with the 'shift' or 'crtl' modifiers, you may define up to 30 text input macros that can be used at the press of a key. This is similar to the functon key macros found on many popular terminal emulation programs.

A '/n' at the end of a function key definition will send a <CR> after the text. The function key definitions also allow placeholders, similar to the Chapter 9 [Operation], page 19.

24 AmIRC's ARexx Port

24.1 Overview

AmIRC supports a versatile ARexx Port.

24.2 Reference

24.2.1 SAY

Usage: `SAY TEXT/F`

DESCRIPTION:

Treats *TEXT* just like it has been entered in the string gadget of the current window.

EXAMPLES:

'`say '/JOIN #Amigager''`'

RELATED TOPICS: See Section 24.2.6 [RX_ECHO], page 63.

24.2.2 ISCONNECTED

Usage: `ISCONNECTED`

DESCRIPTION:

Checks whether a connection is currently established. Returns RC 0 if the link is up, RC 5 otherwise.

EXAMPLES:

'`ISCONNECTED; say rc`'

RELATED TOPICS:

24.2.3 GETMYNICK

Usage: `GETMYNICK`

DESCRIPTION:

Returns your current nickname.

EXAMPLES:

'`GETMYNICK; say result`'

RELATED TOPICS: See Section 24.2.4 [RX_GETSERVERNAME], page 62.

24.2.4 GETSERVERNAME

Usage: `GETSERVERNAME`

DESCRIPTION:

Returns the name of the IRC server you are currently connected to.

EXAMPLES:

'`GETSERVERNAME; say result`'

RELATED TOPICS: See Section 24.2.3 [RX_GETMYNICK], page 62.

24.2.5 USERHOST

Usage: `USERHOST NICK/A`

DESCRIPTION:

Returns the userhost specification belonging to *NICK*.
AMIRC keeps a internal list of the userhosts of every nick that occured yet. You can query this database with this command. Note however that it is possible that the userhost isn't in the database yet; this command will **not**query the server than, but simply return an error.

EXAMPLES:

'`USERHOST Olli; say result`'

RELATED TOPICS: See Section 24.2.3 [RX_GETMYNICK], page 62.

24.2.6 ECHO

Usage: `ECHO P=PREFIX/K, TEXT/F`

Description:

Sends text to the listview

Examples:

'`ECHO P=Kewlja This is echo test`'

24.2.7 RAW

Usage: `RAW TEXT/F`

Description:

Sends text to the server in raw format.

Examples:

'`RAW PRIVMSG phalanx: heya! whats happening?`'

24.2.8 CHANNELS

Usage: `CHANNELS NULL`

Description:

Returns a list of channels currently joined in the format: "#channel1 #channel2 etc."

Examples:

'`options results; channels; say result`'

24.2.9 GETCHANNEL

Usage: `GETCHANNEL`

Description:

Returns the name of the channel currently active in the format: "#channel."

Examples:

'`options results; getchannel; say result`'

Related Topics: See Section 24.2.8 [RX_CHANNELS], page 63.

24.2.10 GETSELECTEDUSER

Usage: `GETSELECTEDUSER`

Description:

Returns the Nick the user highlighted in the userlist window.

Examples:

'`options results; getselecteduser; say result`'

Related Topics: See Section 24.2.11 [RX_GETUSER], page 64, See Section 24.2.12 [RX_GETUSERS], page 64.

24.2.11 GETUSER

Usage `GETUSER /N`

Description:

Returns the nick of the user number given on the active channel userlist.

Examples:

'`options results; getuser 1; say result`'

Related Topics: See Section 24.2.10 [RX_GETSELECTEDUSER], page 64, See Section 24.2.12 [RX_GETUSERS], page 64.

24.2.12 GETUSERS

Usage `GETUSERS`

Description:

Returns a string containing the list of users on the active channel.

EXAMPLES:

'options results; getusers; say result'

RELATED TOPICS: See Section 24.2.11 [RX_GETUSER], page 64, See Section 24.2.10 [RX_GETSELECTEDUSER], page 64.

24.2.13 GETUSERSTATE

Usage `GETUSERSTATE /N`

DESCRIPTION:

Returns the status of user "num" on the active channel.

'RC=0 User is online, but has no special status.'
'RC=1 User is Channel Operator.'
'RC=2 User is online and has been "Voiced".'
'RC=5 User # not in the userlist.'

EXAMPLES:

'options results; getuserstate 1; say rc'

24.2.14 GETLINE

Usage `GETLINE`

DESCRIPTION:

Waits for a line from the server and returns the following stem variables;.

'LINE = The entire, unparsed, line recieved.'
'LINE.PREFIX = The Nick and user@host.'
'LINE.COMMAND= The MSG Type.'
'LINE.DEST = The channel the message was written to.'
'LINE.REST = The message text.'

EXAMPLES:

'options results'

getline

text=line prefix=line.prefix comm=line.command dest=line.dest rest=line.rest say text,prefix,comm,dest,rest

RELATED TOPICS: See Section 24.2.19 [RX_PARSELINE], page 67, See Section 24.2.16 [RX_GETLASTLINE], page 66.

24.2.15 GETLINECOUNT

Usage `GETLINECOUNT`

DESCRIPTION:

Returns the number of lines in the channel buffer for the active window channel.

EXAMPLES:

'`options results; getlinecount; say result`'

RELATED TOPICS: See Section 24.2.14 [RX_GETLINE], page 65.

24.2.16 GETLASTLINE

Usage `GETLASTLINE`

DESCRIPTION:

Returns the last private message received in the format;. :nick!~user@host Messagetype Nick :message text.

EXAMPLES:

'`options results; getlastline; say result`'

RELATED TOPICS: See Section 24.2.19 [RX_PARSELINE], page 67, See Section 24.2.14 [RX_GETLINE], page 65.

24.2.17 GETAWAYSTATE

Usage `GETAWAYSTATE`

DESCRIPTION:

Returns your away status.

'`rc=0 Not away.`'

'rc=5 Away.'

EXAMPLES:

'options results; getawaystate; say rc'

24.2.18 GETVERSION

Usage GETUSERS

DESCRIPTION:

Returns the Version number of AmIRC.

EXAMPLES:

'options results; getversion; say result'

24.2.19 PARSELINE

Usage PARSELINE

DESCRIPTION:

Parses the line specified and returns the following stem variables;

'LINE = The entire, unparsed, line recieved.'
'LINE.PREFIX = The Nick and user@host'
'LINE.COMMAND= The MSG Type'
'LINE.DEST = The channel the message was written to.'
'LINE.REST = The message text.'

EXAMPLES:

'options results'

getlastline last=result parseline last

text=line prefix=line.prefix comm=line.command dest=line.dest rest=line.rest

say text,prefix,comm,dest,rest

RELATED TOPICS: See Section 24.2.14 [RX_GETLINE], page 65, See Section 24.2.16 [RX_GETLASTLINE], page 66.

25 Magic User Interface

25.1 About MUI

This application uses

MUI - MagicUserInterface

`'Without MUI, AmIRC wouldn't exist.'`

If you want to register directly, feel free to send

DM 30.- or US$ 20.-

to

Stefan Stuntz
Eduard-Spranger-Strae 7
80935 Mnchen
GERMANY

25.2 Why MUI?

Why MUI? you ask...

MUI is the most powerful application framework toolkit available for any platform. MUI allows applications to provide their users with a highly usable, fast and configurable user interface which works well on any screen setting, from a 640x200 NTSC display to a 1280x1024 true color graphics board. MUI applications use less memory than comparable applications since they can rely on a huge set of shared class code, which is especially true if several MUI apps are run on the same machine.

Plus, MUI's powerful OO class methods allows the application programmer to work on the program's functionality instead of doing repetitive low level tasks, resulting in much more powerful applications given the same development times. MUI's GUI classes overcome the severe design bugs in the ROM internal BOOPSI gadget classes, allowing for complex GUI objects without high priority CPU hogging or potential system deadlocks.

MUI compacts over 4 years of development and optimization in a powerful toolkit, an effort no application programmer can afford to invest in his own development.

26 Those Deserving Favour

26.1 Thanks to::

The author wishes to thank the following people for their help during AmIRC development:

'Karl 'Phalanx' Bellve'
: Who provided space, connectivity and cpu time for running the AmIRC support area. He also helped during development, and did alot of encouragement. :-)

'Laurence 'DuRaN' Walton'
: Who did write the documentation, first installer and also helped alot during development in other respects.

'Christopher 'ChrisDi' Dietz'
: Who did the German catalog translation.

'Philippe 'PhB' Brand'
: Who did the french catalog translation.

'Fredrik Rambris'
: Who did the swedish catalog translation.

'Pascal 'arcade' Eeftinck'
: Who did the dutch catalog translation.

'Juan 'Goliath' Ramirez'
: Who did the Spanish translation

'Manuel 'CCCP' Giorgini'
: Who did the Italian translation

'Troy 'Doogie' Davis'
: Who helped much with testing and developing the ARexx port, besides reporting bugs and making useful suggestions of other kinds.

'Robert 'RobR' Reiswig'
: Who did write the installer script.

'Mike 'Mike' Fitzgerald'
: for writing the ARexx Section of the docs

'Olaf 'Olsen' Barthel'
'Michael 'mlelstv' van Elst'
'Kenneth 'Kenneth' Jennings'
'David 'Deathland' Zvekic'
: for their help with debugging the as225r2 code.

'Klaus 'kmel' Melchior'
: for his help with tracking down several MLink problems.

'Andreas M. 'bonzo' Kirchwitz'
: for useful advice and showing a good load of humor during the //KICKBAN test.

'Ralph 'rbabel' Babel'
: who helped to track down a nasty bug in slink's constructor priority ordering.

'The SAS/C Amiga Development team'
'(Especially Steve Krueger)'
: For their continued support of the SAS/C Compiler

'Holger 'hkr' Kruse'
: who pointed out the easy solution to the *WTF is my own IP number?* problem

'Jason 'morden' Murray'
'Robert 'Devo' Bradman'
: who drew the OP and VOICE icons

'Angela 'Nessy' Schmidt'
'Christoph 'Pepo' Feck'
'Sascha 'Master' Lentz'
: who helped test several AmIRC features in a real-life environment (say #AmigaGER ;-)

'The IRC admin at irc.uni-paderborn.de'
'(Thomas 'Heager' Thissen)'
: which served as the primary test site

'Brian 'SmknDHerb' Maurino'
'Sandro 'Sandro' Catallo'
'Robert 'Gostrider' Wilkins'
: Who created and compiled the demo sounds

'Fionn 'Fionn' Behrens'
'Frank 'Daddie' Peters'
'Holger 'Heureka' Jakob'
'Jens 'Jens' Hnerberg'
'Mathias 'Merlin' Mischler'
'Michael 'mickh' Hohmann'
'Nicole 'Talyn' Brandt'
'Peter 'Stoebi' Schulz'
'Roy 'AK' Millican'
'Thomas 'logo' Butter'
: for early testing and design suggestions

'Christoph 'CeeGee' Glicher'
'Danny 'Danny' Amor'
'Markus 'Ill' Illenseer'
'Markus 'Lammy' Lamers'
'Matthias 'DrMabuse' Lhr'
: who simply bugged me to get on the list :)

'Osma 'Tau' Ahvenlampi'
: The major inspiration for the IGNORE support in AmIRC

This Document Prepared by Laurence Walton

27 How to Report Bugs

27.1 Bug Reports

Please send bug reports to '`AmIRC@vapor.com`', not to my standard mailbox. Include in the report the OS version of your amiga, The version of MUI that you have installed, and the version of AmIRC that you are using.

Problems or additions of the AmIRC documentation should be directed to:

'`lww@vapor.com`'

28 Problems, Causes, & Cures

28.1 Known Bugs

- You absolutely MUST install SetPatch 40.16 (or higher), otherwise you will stumble over the broken pool functions in 3.0/3.1 exec
- The main window sometimes doesn't refresh properly when SIMPLE_REFRESH is used. Known bug in MUI 2.3's listview refresh sometimes throwing away damage regions. Fixed in MUI 3.
- System deadlock when having an ObtainGIRPort hack installed. This hacks break a few of the ROM internal BOOPSI gadget classes (notable the propgclass which is used by AmIRC for it's scroller) and MUST not be used. Older versions suggested to install such a hack to avoid MagicMenu lockups; this was before the system hostility of the ObtainGIRPort hack was unveiled.
- There are various problems when using AmIRC as a CLChat (ArcNet) client. These are all caused by bugs within CLChatIRCD and can't be fixed on AmIRC's side.
- ALT-CURSOR-UP/DOWN doesn't work with MultiCX. MultiCX's "advanced Edithook" will eat Cursor-UP/Down-Inputevent on it's own so they can't be processed by AmIRC. Use a different edit hook.
- ZFREQ causes AmIRC to fail upon start. ZFREQ seems to be unable to bear an application allocating a real huge number of ASL_Filerequest structures. Use a different file requester patch.
- Neither HILITE nor URL-Grabber work on text written by yourself. This is due to internal reasons, I'm not sure if it is worth changing.

29 How to Get Help & Updates

29.1 Support

'AmIRC Support Mailing List'

To keep up to date on AmIRC improvements or to get answers to AmIRC problems, Join the AmIRC support mailing list by sending email to:

<AmIRC-Request@vapor.com>

with the text "ADD" in the body.

'AmIRC FTP Support Site'

For AmIRC Updates & Patches, ftp to:

ftp.vapor.com, /pub/amirc/

'AmIRC Support on the World Wide Web'

Or, Alternatively, Visit the AmIRC WWW Support Site at:

http://www.vapor.com/amirc/

Index

(Index is nonexistent)

Table of Contents

1 Copyright **1**
- 1.1 Copyright 1

2 Registering AmIRC **3**
- 2.1 License 3
- 2.2 Registration 3

3 What is IRC, by the way? **5**
- 3.1 Introduction 5

4 Terms You Might Encounter **7**
- 4.1 IRC Terms 7

5 Things that make AmIRC special! **11**
- 5.1 Features 11

6 What is needed to run AmIRC? **13**
- 6.1 Hardware 13
- 6.2 Software 13

7 Getting AmIRC to run **15**
- 7.1 Requirements 15

8 Connecting to a server **17**
- 8.1 Setup 17

9 How Does AmIRC Work? **19**
- 9.1 Using the Interface 19

10 Pull-Down Menus **21**
- 10.1 Reference 21
  - 10.1.1 AmIRC Menu 21
  - 10.1.2 Windows Menu 21
  - 10.1.3 User Modes Menu 22
  - 10.1.4 Settings Menu 23
  - 10.1.5 Rexx Menu 24

11 Sending commands to other people **25**
- 11.1 Introduction 25
  - 11.1.1 CTCP or *Client To Client Protocol* 25
  - 11.1.2 CTCP Commands 25

12 Direct client connections **27**
- 12.1 What is DCC? 27
- 12.2 DCC File Transfers 27
  - 12.2.1 DCC Send 27
  - 12.2.2 DCC Move 27

12.2.3 DCC Get 27
12.3 DCC Chat 28

13 Commands 29

13.1 Overview 29
13.2 Reference 29
13.2.1 AWAY 29
13.2.2 INVITE 30
13.2.3 ISON 30
13.2.4 JOIN 30
13.2.5 KICK 31
13.2.6 LASTLOG 31
13.2.7 LEAVE 31
13.2.8 LIST 33
13.2.9 Logging Options 33
13.2.10 Channel MODE 33
13.2.11 User MODE 34
13.2.12 MSG (PRIVMSG) 34
13.2.13 Names 35
13.2.14 NICK 35
13.2.15 NOTICE 35
13.2.16 QUIT 36
13.2.17 RAW 36
13.2.18 TIME 37
13.2.19 TOPIC 37
13.2.20 USERS 37
13.2.21 USERHOST 37
13.2.22 VERSION 38
13.2.23 WHO 38
13.2.24 WHOIS 38
13.2.25 WHOWAS 39

14 Customizing your environment 41

14.1 Overview 41

15 Changing MUI Settings 43

15.1 MUI Preferences 43

16 Changing AmIRC Options 45

16.1 AmIRC Setup 45

17 Messages Page 47

17.1 Misc Messages 47
17.2 Quit Messages 47
17.3 AWAY Messages 47

18 Actions Page 49

18.1 Actions 49

19 Events Page 51

19.1 Events 51
19.1.1 Columns 51
19.1.2 Events 51

20 GUI Configuration . . . 53

20.1 GUI . . . 53
20.1.1 Userlist: . . . 53
20.1.2 Misc . . . 53
20.1.3 Text Colors . . . 53
20.1.4 Logging . . . 54

21 Misc Configuration . . . 55

21.1 Miscellaneous Configuration Items . . . 55
21.1.1 CTCP COUND handling . . . 55

22 "Lists..." Menu Item . . . 57

22.1 Lists Configuration . . . 57
22.1.1 Alias Tab . . . 57
22.1.2 Ignore Tab . . . 57
22.1.3 Hilite Tab . . . 57
22.1.4 Pattern Definitions: . . . 57

23 Function Key Definitions . . . 59

23.1 Function Keys . . . 59

24 AmIRC's ARexx Port . . . 61

24.1 Overview . . . 61
24.2 Reference . . . 61
24.2.1 SAY . . . 61
24.2.2 ISCONNECTED . . . 61
24.2.3 GETMYNICK . . . 62
24.2.4 GETSERVERNAME . . . 62
24.2.5 USERHOST . . . 62
24.2.6 ECHO . . . 63
24.2.7 RAW . . . 63
24.2.8 CHANNELS . . . 63
24.2.9 GETCHANNEL . . . 63
24.2.10 GETSELECTEDUSER . . . 64
24.2.11 GETUSER . . . 64
24.2.12 GETUSERS . . . 64
24.2.13 GETUSERSTATE . . . 65
24.2.14 GETLINE . . . 65
24.2.15 GETLINECOUNT . . . 66
24.2.16 GETLASTLINE . . . 66
24.2.17 GETAWAYSTATE . . . 66
24.2.18 GETVERSION . . . 67
24.2.19 PARSELINE . . . 67

25 Magic User Interface . . . 69

25.1 About MUI . . . 69
25.2 Why MUI? . . . 69

26 Those Deserving Favour . . . 71

26.1 Thanks to:: . . . 71

27 How to Report Bugs . . . 73

27.1 Bug Reports . . . 73

28 Problems, Causes, & Cures **75**
28.1 Known Bugs 75
29 How to Get Help & Updates **77**
29.1 Support 77
Index **79**