AmIRC

COLLABORATORS

	TITLE : AmIRC		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		July 25, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

AmIRC

1.1 AmIRC.guide

AmIRC
*****

Welcome to AmIRC, the AmigaOS IRC client.

- Introduction -

- Reference section -

- Other Stuff: -

1.2 AmIRC.guide/Copyright

Copyright
****

Copyright
=========

AmIRC has been designed, developed and written by

Oliver Wagner
Landsberge 5
45549 Sprockhövel
Germany
E-Mail: (owagner@lsd.wupper.de)

All Rights Reserved.

This document compiled by Laurence Walton (lww@pgh.net)
©1996 Oliver Wagner
All Rights Reserved

Installer Script
©1995,1996 Robert C. Reiswig
All Rights Reserved

No part of this archive may be redistributed except with the entire contents of the original archive.

1.3 AmIRC.guide/Registration

Registering AmIRC
****

License
=======

AmIRC 1.0 Registered Software License Agreement

AmIRC 1.0 is Copyright © 1996 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 1.0 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 1.0 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.

Registration
============

If you have not registered for AmIRC, it will prompt you to do so upon startup. Registration at the US site requires $25.00 in US funds for an emailed keyfile, or $30.00 for a disk mailed via postal service. The Germany Site requires 35,-DM for an emailed keyfile and 40,-DM for a floppy sent via postal service.

1.4 AmIRC.guide/IRC

What is IRC, by the way?
****

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.

#amiga
======

If you are reading this, you are most likely using one of the Amiga family of computers (if you are not, BUY ONE! :), and will likely want to check out #amiga on IRC. #amiga is one of the most populated channels of IRC, with an average of 30 to 60 users on at any given time. #amiga is often visited by amiga hardware and software developers, and is sometimes a forum for formal discussions held by dignitaries from software/hardware development companies.

#amiga Etiquette

There is a sort of an unwritten code of conduct for IRC, and especially #amiga. This section is included so that you dont find yourself /kicked or /banned and dont understand why.

```
DO's
----

DO:

     Enjoy your visit to #amiga

     Offer to help someone IF you know what your talking about.

     Be polite to other users.

DON'T's
-------

DON'T:

     Ask questions about software/hardware problems without reading the
     manual first (RTFM!).

     Ask a question more than once. If someone can help you, they will.

     Ask for latest amiga news or info on new software, see newsgroup
     comp.sys.amiga.announce for this.

     Ask for OPs.  You won't get it.

     Pound or slam Amigas.  If you don't like them, don't /join #amiga.

     Pound or slam Amiga Software (the coders might be there :)

     Offer to help a user with a problem if you don't understand it
     yourself.

     Offer or ask for commercial software or the keys to activate
     demos.

     Plug developers who appear from time to time with questions and bug
     reports, use the proper channels for this (i/e, the developer's support
     address).  Note that you might ask first if they are willing to discuss
     problems/bugs with their product.
```

1.5 AmIRC.guide/IRC Terms

```
Terms You Might Encounter
*************************

IRC Terms
=========

#amiga
     IRC Channel on Efnet and Undernet for users interested in the
     Amiga family of Computers.
```

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, #amiga), 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.

Command
A message to an IRC server or to a local or remote client. Commands generally begin with a /. See Commands.

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 CTCP.

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 DCC.

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.wupper.de
|      |       |        |      |_ Domain
|      |       |        |________ Network
|      |       |_________________ Machine
|      |_________________________ Login
|________________________________ Nick
```

Voice

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

1.6 AmIRC.guide/Features

Things that make AmIRC special!
****

Features
========

- runs with both AmiTCP and AS225 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.

- 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 ":".

* 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 copying URLs to clipboard.

* netsplit detection.

1.7 AmIRC.guide/Requirements

What is needed to run AmIRC?
****

Hardware
========

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

Software
========

AmIRC requires MUI v.3.1 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 3.1 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 MUI 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 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'.

1.8 AmIRC.guide/Installation

Getting AmIRC to run
****

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 requirements for the optimized version of AmIRC, delete the AmIRC file and rename "AmIRC020" as "AmIRC".

1.9 AmIRC.guide/Connecting

Connecting to a server
****

Setup
=====

Once you have installed 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. The Delete Server Entry gadget will remove a server entry that you no longer wish to use.

1.10 AmIRC.guide/Operation

How Does AmIRC Work?
****

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 configuration window, accessable by selecting 'setup...' from the AmIRC settings 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.

1.11 AmIRC.guide/Menus

Pull-Down Menus
****

Reference
=========

AmIRC	AmIRC menu
Windows	Windows menu
User Modes	User Modes menu
Settings	Settings menu
Rexx	Rexx menu

1.12 AmIRC.guide/AmIRC

AmIRC Menu

Select Server...
Opens a window from which you can select various options for connecting to an IRC server. See Connecting.

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...
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.

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

1.13 AmIRC.guide/Windows

```
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 her 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 DCC Section 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 DCC.
```

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 DCC.

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.

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.

1.14 AmIRC.guide/User Modes

User Modes Menu

Invisible?

Toggles your client between invisible or not invisible. See IRC Terms.

Receive Wallops?

Toggles this mode on/off. See IRC Terms.

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 Events page of the settings requester) for certain events.
```

1.15 AmIRC.guide/Settings

```
Settings Menu
-------------

Setup...
     Calls the Setup Requester, allowing you to change the many
     settings of AmIRC to suit your needs.

Lists...
     Calls up the Lists Requester, allowing you to configure AmIRC's
     Alias, Ignore, and Hilite lists.

Function Keys...
     Calls the Function Key settings requester, 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.

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

1.16 AmIRC.guide/Rexx

```
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.
```

1.17 AmIRC.guide/CTCP

```
Sending commands to other people
********************************

Introduction
============

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 /DCC sends a request
via CTCP to the receiving client, asking to initiate a direct
connection.

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.

1.18 AmIRC.guide/DCC

Direct client connections
****

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.

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.

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.

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.

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.

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.

1.19 AmIRC.guide/Commands

Commands

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 DCC 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.

Reference

AWAY	Setting an away message
INVITE	Invite a user to a channel
ISON	If a user is on IRC
JOIN	Join (or create) a channel
KICK	Remove a user from a channel
LEAVE	Leave a channel
LIST	List Channels and their topics
LOG	Start Log of Chat Window
MODE	Change a channel mode
MSG	Send a private message (privmsg)

```
  NAMES         Getting a list of names
  NICK          Change your nickname
  NOTICE        Send a notice to a user
  QUIT          End an IRC session
  RAW           Send unprocessed string to the server
  TIME          Query the current time of day
  TOPIC         View or change a channel's topic
  USERS         Listing users on a server
  USERHOST      Getting USERHOST of a user
  VERSION       Query the version of the server software
  WHO           Global user pattern matching
  WHOIS         Get information about a user
  WHOWAS        Get information on a nick no longer logged in
```

1.20 AmIRC.guide/AWAY

```
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 /MSG 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.
```

1.21 AmIRC.guide/INVITE

```
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 invite only(+i),
the user performing the command must be a channel operator for that
channel.

EXAMPLES:

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

1.22 AmIRC.guide/ISON

```
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.
```

1.23 AmIRC.guide/JOIN

```
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,#Amiga
     Join channels #AmigaGER and #Amiga

/JOIN #AmigaGER deutsch
     Join channel #AmigaGER with keyword deutsch.
```

RELATED TOPICS:

See LEAVE.

1.24 AmIRC.guide/KICK

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 #amiga Freddie Take it outside!
     forces 'Freddie' from channel #amiga with the reason 'take it
     outside'.
```

1.25 AmIRC.guide/LEAVE

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 JOIN.

1.26 AmIRC.guide/LIST

```
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 #amiga,#amigaGER
     List channels #amiga and #amigaGER
```

1.27 AmIRC.guide/LOG

```
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.
```

1.28 AmIRC.guide/MODE

```
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 #amiga +tm
     Sets topic protection and makes channel moderated for #amiga.

/MODE #amiga -o Sarek
     Removes channel operator status from user 'Sarek' on channel
     #amiga.

/MODE #amiga +l 50
     Sets limit of 50 users on #amiga.

/MODE #amiga -l
     Removes the limit form #amiga

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.
```

1.29 AmIRC.guide/MSG

```
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 /DCC Chat, 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.
```

1.30 AmIRC.guide/NAMES

```
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 #amiga,#amigaGER
     Will list all users on channels #amiga and #amigaGER, provided the
     channels are visible to the user.

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

1.31 AmIRC.guide/NICK

```
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'
```

1.32 AmIRC.guide/NOTICE

```
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 /MSG for syntax examples.
```

1.33 AmIRC.guide/QUIT

```
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
```

1.34 AmIRC.guide/RAW

```
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’.

1.35 AmIRC.guide/TIME

```
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’
```

1.36 AmIRC.guide/TOPIC

```
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 Channel Modes.

EXAMPLES:

/TOPIC #amiga
     Will display the current topic on #amiga.

/TOPIC #amiga Amigas Rule!
     Will replace the current topic of #amiga with the topic ’Amigas
     Rule’
```

1.37 AmIRC.guide/USERS

```
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'.
```

1.38 AmIRC.guide/USERHOST

```
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
```

1.39 AmIRC.guide/VERSION

```
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'.
```

1.40 AmIRC.guide/WHO

```
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
```

1.41 AmIRC.guide/WHOIS

```
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
```

1.42 AmIRC.guide/WHOWAS

```
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'.
```

1.43 AmIRC.guide/Configuration

Customizing your environment
****

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.

MUI Preferences	Changing MUI Settings
AmIRC Setup	Changing AmIRC Options
Function Keys	AmIRC Functon Key Definitions

1.44 AmIRC.guide/MUI Preferences

Changing MUI Settings
****

MUI Preferences
===============

Most of the configuration dealing with how AmIRC's GUI appears can be customized with MUI. 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.

1.45 AmIRC.guide/AmIRC Setup

Changing AmIRC Options
****

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.

Messages	Messages Tab
Actions	Actions Tab
Events	Events Tab
GUI	GUI Tab
Lists	Lists Menu

1.46 AmIRC.guide/Messages

Messages Page

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.

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.

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'.

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

1.47 AmIRC.guide/Actions

Actions Page
****

Actions
=======

The 12 text gadgets in the 'Label:' column here correspond to the 12 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.

1.48 AmIRC.guide/Events

Events Page
****

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.

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.

Vol
     This sets the volume for the sound sample.  The valid range for
     the volume is 0 - 64.  This is a slider gadget.

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

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’

1.49 AmIRC.guide/GUI

GUI Configuration
****

GUI
===

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

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 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.

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.

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
Highlighted text is not yet implemented in 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.

Logging

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

1.50 AmIRC.guide/Lists

"Lists..." Menu Item
****

Lists Configuration
===================

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.

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.

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.

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.
```

1.51 AmIRC.guide/Function Keys

```
Function Key Definitions
************************

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 user buttons.

1.52 AmIRC.guide/ARexx

AmIRC’s ARexx Port
****

Overview
========

AmIRC supports a versatile ARexx Port.

Reference
=========

SAY	Treat something just like it was entered in the input string.
GETMYNICK	Request current nick name.
ISCONNECTED	Check whether connection is established.
GETSERVERNAME	Request current server name.
USERHOST	Query Userhost of a nick.
ECHO	Echos text to the current listview.
RAW	Sends raw text to the server.
CHANNELS	Returns string showing joined channels.
GETCHANNEL	
GETSELECTEDUSER	
GETUSER	
GETUSERS	
GETUSERSTATE	
GETLINE	
GETLINECOUNT	
GETLASTLINE	
GETAWAYSTATE	
GETVERSION	
PARSELINE	

1.53 AmIRC.guide/RX_SAY

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:
```

1.54 AmIRC.guide/RX_ISCONNECTED

```
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:
```

1.55 AmIRC.guide/RX_GETMYNICK

```
GETMYNICK
---------

Usage: GETMYNICK

DESCRIPTION:

Returns your current nickname.

EXAMPLES:

GETMYNICK; say result

RELATED TOPICS: See RX_GETSERVERNAME.
```

1.56 AmIRC.guide/RX_GETSERVERNAME

```
GETSERVERNAME
-------------
```

```
Usage: GETSERVERNAME

DESCRIPTION:

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

EXAMPLES:

GETSERVERNAME; say result

RELATED TOPICS: See RX_GETMYNICK.
```

1.57 AmIRC.guide/RX_USERHOST

```
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 RX_GETMYNICK.
```

1.58 AmIRC.guide/RX_ECHO

```
ECHO
----

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

DESCRIPTION:

Sends text to the listview

EXAMPLES:

ECHO P=Kewlja This is echo test
```

1.59 AmIRC.guide/RX_RAW

```
RAW
---

Usage: RAW TEXT/F

DESCRIPTION:

Sends text to the server in raw format.

EXAMPLES:

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

1.60 AmIRC.guide/RX_CHANNELS

```
CHANNELS
--------

Usage: CHANNELS NULL

DESCRIPTION:

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

EXAMPLES:

options results; channels; say result
```

1.61 AmIRC.guide/RX_GETCHANNEL

```
GETCHANNEL
----------
```

1.62 AmIRC.guide/RX_GETSELECTEDUSER

```
GETSELECTEDUSER
---------------
```

1.63 AmIRC.guide/RX_GETUSER

```
GETUSER
-------
```

1.64 AmIRC.guide/RX_GETUSERS

```
GETUSERS
--------
```

1.65 AmIRC.guide/RX_GETUSERSTATE

```
GETUSERSTATE
------------
```

1.66 AmIRC.guide/RX_GETLINE

```
GETLINE
-------
```

1.67 AmIRC.guide/RX_GETLINECOUNT

```
GETLINECOUNT
------------
```

1.68 AmIRC.guide/RX_GETLASTLINE

```
GETLASTLINE
-----------
```

1.69 AmIRC.guide/RX_GETAWAYSTATE

```
GETAWAYSTATE
------------
```

1.70 AmIRC.guide/RX_GETVERSION

```
GETVERSION
----------
```

1.71 AmIRC.guide/RX_PARSELINE

```
PARSELINE
---------
```

1.72 AmIRC.guide/About MUI

```
Magic User Interface
********************

About MUI
=========

                               This application uses


                             MUI - MagicUserInterface

                     (c) Copyright 1993/94 by Stefan Stuntz


                        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-Straße 7
                                 80935 München
                                    GERMANY

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.

1.73 AmIRC.guide/Revision History

Developments in AmIRC
****

Revision History
================

0.87beta

- list class now avoids "plane optimization" if AmIRC runs on a cybergfx chunky mode screen. - made the userlist scroller less jumpy. - fixed a few boolean config settings which weren't properly saved. - fixed screen refresh problems with mark selecting and scrolling. - during config save, "," inside the autojoin channel settings weren't properly escaped, causing the server list to be messed up. - AWAY state will now be visualized by rendering the nick underlined with FILLPEN. - it's now possible to measure the user-server round trip time by issuing the /PING command without parameters. - double clicking on an entry in the URL grabber window will now cause the entry to be copied to the clipboard.

0.86beta

- fixed several problems with listview refresh. - cut&paste marking is a bit less flashier now again. - function keys now allow for the same placeholders as user op buttons. - function key assignments ending with a "\n" now cause the input to be directly processed as if the user pressed ENTER manually. - alias definitions now work even if they start with a "/".

0.85beta

- sounds and userlist settings weren't updated upon loading the config file. - fixed TAB messaging to properly cycle in both directions. - fixed several bugs regarding cut&paste marking. - reduced screen refresh during cut&paste.

```
0.84beta
-------
- now requires at least MUI 3.1 to work.  - replaced the main text list
with a custom list object optimized for   this particular purpose. This
results in about 3 times faster   scrolling and generally less CPU
usage during text output. Also,   the new list class is much more
memory efficient and works better   during relayout. Note that the
speed isn't fully optimized yet.  - the main text listview now supports
mark/cut to clipboard.  - fixed the desynch problems with the ARexx
GETLINE command   which could cause lines to be missed. Now, GETLINE
will buffer   the last 50 server lines internally and will use a special
 counter variable bound to your ARexx script to make sure you
receive lines in order.  - rewrote the internal prefs handling
completely. Decreased AmIRC's   memory usage by a few 100k's since now
all prefs BOOPSI objects   are created dynamically.  - AmIRC now saves
a textual configuration file for easy editing.  - the auto prefs saving
during exit can now be turned off with   a menu option.  - the prefs
requesters now have "Use" and "Cancel" buttons.  - event sounds are no
more loaded twice or more if the same sounds   are specified for
different events.  - fixed NOTIFY to no longer turn recognized nicks
into upper case   in the list.  - TAB now works more like the ircII tab
script; cycling through the   list of nicks who last send messages to
you. SHIFT-TAB cycles forward.  - the "," nick destination will now
also by set by incoming DCC   CHAT messages.  - "Connect Thread" didn't
work when the executable path contained   spaces (i.e. "Ram Disk:" ;) -
real long channel names are now cutted before displayed in the
channel window and window menu.  - the DCC messages generated by DCC
sub threads didn't use the   color setup for DCC messages.

0.83beta
-------
[internal only]

0.82beta
-------
- changed kickban window layout to fit on medres screens - fixed a
nasty internal bug which caused innocent memory to be overwritten
during preference save. Could cause nifty side effects under certain
circumstances.  - changed yet another bug in the prefs window setup
which caused a   bad crash when /IGNORE, /HILITE or /ALIAS were used
after the list   window was opened once.  - reduced overall stack usage
and default stack size to 24k.  - changed several string buffers to fit
into boopsi's stringclass   shared buffer.  - several attempts to
reduce overall memory usage, esspecially   using new MUI 3.2 features.

0.81beta
-------
- startup failure caused a FreeSignal(0) to happen.  - added new
OP/VOICE icons which should look better on a 4 color screen.  - several
GUI fixes.  - reduced signal bit usage to avoid trouble with certain
inet225 versions.  - added NORESETHANDLER Tooltype/Shell switch.  -
finally changed DCC host ip lookup to use getsockname() on the server
socket; this should fix ALL kind of ENV:HOSTNAMEish trouble. TIA
users still must set TIADCC or change to a real IP provider :) - fixed
a severe bug in the DCC CHAT IPC system which caused a   non-protected
access to a pool from several processes, possibly   resulting in a
nifty crash.  - fixed a few potential crashes when moving objects while
```

dynamically creating prefs windows. - event rexx script parameter extension caused a nice crash with very long messages. - changed behavior of the "DCC Receive Dir" field. From now on, changes in this path will also affect already queued DCC receive requests which haven’t be renamed to a different filename and path before. This is the "intuitive" behavior. - added "borderright" tooltype/shell option to have the window size border moved to the right. This should really be an MUI prefs option.

0.80beta

- Wallops-Display will now display the source of the Wallops message - fixed a crash when selecting "Sound?" in events menu. - added fixed catalog translations. - the default initialisation of the server list was messed up and caused both enforcer hits and junk in the password entries. - sound volume settings get messed up, causing sounds to be played with volume 0 until the sound options window was opened once.

0.79beta

- severe bug in the as225 part of the tcp code caused a AmiTCP LVO to be called on SockBase, resulting in messed up results up to mere crashes. - prefs setup modification caused the sound dt support to not work properly. - prefs setup modification caused /LOG to crash immediately. - did various GUI fixes.

0.78beta

- added GETUSERS command which returns a string of all users on a channel of the form "nick1 nick2 nick3"... - %p replacement within EVENT ARexx calls didn’t work at all. - some internal & GUI cleanup. - changed DCC port allocation scheme to avoid "reusing" active but delayed DCC requests with different users and to avoid "No socket" errors with MLink. - workaround a few MLink close() bugs in DCC send. - now accepts 422 response as End-Of-MOTD. Fixes problems with IRC servers which started up with "Error MOTD file is missing", causing login, autojoin etc.pp. handling to not work. - topics weren’t properly cleared when hopping servers or channels. - rename of DCC-CHAT connections upon NICK didn’t work properly. - "Connect Thread" didn’t work when no autojoin was specified for the selected server (new instance failed immediately with argument parse error) - private messages output caused %-format-codes to be interpreted instead of being ignored. - "/MSG ," addressed the last PRIVMSG source even if it was a PRIVMSG of an ignored user. - now all ALT-SPACE inside input strings are replaced by simple spaces, to avoid trouble with IRC server parsing. - reduced overall memory usage by making several of the prefs windows dynamic. - added Language Tooltype/Shell-Option to manually select a language catalog, overriding locale prefs. - fixed several GUI quirks.

0.77beta

- fixed a potential WaitForEver() problem in DCC Receive threads. - changed user mode parsing to better suit localisation. - event rexx scripts can now take an optional parameter %p which will be replaced with the unparsed server line which caused the event to happen. - added "PARSELINE" ARexx command which will split it’s argument line into

LINE.PREFIX, LINE.DEST, LINE.REST and LINE.COMMAND just like GETLINE does. - ARexx-Cmd "ISCONNECTED" didn't work at all, but returned 0 always. - added GETAWAYSTATE Arexx command. - fixed HILITE not working properly if the matching word was at the   end of a line. - abuse of MUI's label objects caused several dynamic texts to react oddly to "_" within them (most notable the kickban win title).

0.76beta

- some tangling Printf() debug statement in the DCC Receive handler caused   enforcer hits and severe crashes when several DCC Receive threads were   active. - added GETCHANNEL, GETSELECTEDUSER, GETUSER and GETUSERSTATE ARexx commands.   All take an optional WINNUM argument, GETUSER and GETUSERSTATE require an   user index. GETUSER returns the name of the indexed user of the current window, GETUSERSTATE returns RC=1 if the user is OP, RC=2 if he has voice. - added GETVERSION ARexx command to check AmIRC's version. - added GETLASTLINE, GETLINECOUNT and GETLINE arexx commands. GETLINE   will wait(!) for a new line from the server to arrive and return this   line into the "LINE" variable. It will also return the parsed line components into LINE.PREFIX (userhost), LINE.COMMAND (either numerical or alpha command), LINE.DEST and LINE.REST. The meaning of the components   depends on the specific message. - added new /RXRETURN user command. This is merely to support scripts   and ought to be used with /ALIAS. It will return it's first and following   parameters with the current GETLINE command in LINE.COMMAND and LINE.REST   respectivly. - fixed several mode changes command templates which were messed up. - removed auto turn off of input filtering in string gadgets again. Will now internally accept CTRL-V, CTRL-U (!), CTRL-B, CTRL-G and CTRL-W for entering formatting codes into strings, regardless of the IControl input filtering settings. - lines which contained only control codes messed up the word wrap   engine. - fixed a stupid bug in notify which caused ISON request to be not   send properly under certain circumstances.

0.75beta

- /LIST with a single channel as an argument will no longer cause a complete   list to be requested. - fixed "repeat-on-cr-pressed" bug in the string gadgets. - added parsing of 271/272 response codes (Undernet SILENCE list). - windows are properly closed upon close request; no longer waits for the   PART message from the IRC server. - vbar rendering in the main listview will now completely be done by MUI for efficency and exact positioning. - added netsplit detection. - string gadget has now turned off input filtering by default. - string gadget accepts up to 1024 chars and splits accordingly

0.74alpha

- some MUI3 dependancy slipped in the window creation function.

0.73alpha

- pressing TAB returned garbage if no user was referenced in the user listview. - the first window didn't properly show the nickname. - load and save config with file requester failed badly due to misinitalized   screen variable. - added RAW, ECHO and CHANNELS arexx commands. - fixed several bugs with cycle chains and keyboard control of gadgets. - raised stack size for DCC CHAT threads. Should fix all kinds of crashes, esspecially with as225/inet. - sound loading was severly messed up and caused about EVENTS^2 datatypes load to happen. - fixed WHO layout a bit. - extended the /LIST command to properly work in "shell mode" with pattern and -MIN arguments.

0.72alpha

- switched back to using "GetHostId()" if ENV:HOSTNAME doesn't exist or is invalid. Should fix all kind of DCC trouble with MLink. Make sure that ENV:HOSTNAME is either correct or not existant if you use MLink. - fixed a few typos and IRC response parsing problems. - Topics longer than 127 characters caused a real bad crash. - reduced memory usage a bit. - %p in alias expansion with missing arguments caused garbage to be inserted. - fixed some bugs in the listview setup. - added basic log capability (no filtering yet, only RAW output). - URL grabber messed up news: URLs - seperate user list windows can now be configured to start up opened or closed. - will now properly rename query windows and dcc chat sessions upon receiving of a NICK change message. - misc. fixes.

0.71alpha

- something in the optimizer seems to have broken as225r2 compatibility. Now compiled without optimizer again. Subject for further investigation. - didn't work with MUI 2.x due to use of popscreen.mui. Will now display a simple string gadget then.

0.70alpha

- changed internal string table layout; internal string access is now much faster, even when strings are taken from a catalog file. - /DCC LIST now works. - /WALLOPS was broken. - Query windows can now be opened on a different public screen. - DCC requests are now also accepted when sent with a "NOTICE" to accomodate for certain buggy DCC implementations. - In non-alphasort mode userlist entries were added to the top, not to the bottom of the list.

0.69alpha (quickfix)

- subitems for the window sub menu weren't no longer added, making closing windows crash badly.

0.68alpha

- CTCP PING didn't work. - holding down ENTER caused the same text to be continously send. - SHIFT-TAB will now insert "Nick: " into the input line. - CTRL-UP/DOWN and ALT-UP/DOWN inside a query window caused instant crash. - String class was, in cooperation with some MUI bug, loosing signal bits. This caused frame gadgets to not work any longer. - The bubble help of the userlist action buttons will now display the real command assigned to the button. - Userlist action buttons now work if no user is selected and no %u was specified. - /ME within dialog windows will now work just like /DESCRIBE.

0.67alpha

```
--------
- DCC windows now use tiny fonts for the lists.  - optimized TextFit()
wasn't working correctly, cutting of   characters depending on the font.
- enforcer hits with commands which had a single channel name   as a
parameter ("/NAMES #Amigager" for example).  - /RX will now find
commands in PROGDIR:Rexx - Rexx-on-Event didn't work at all - added a
few more sound samples by Robert Wilkens - the event manager now has
"Test sound" buttons - Connect and Autodownload windows shouldn't cause
trouble any more   (recycled MUIMasterBase between tasks :) - the "is
this mail or news" guess in the URL grabber   always assumed mail

0.66alpha
--------
- userlist action button rows which contain no labels will now be
hidden - the first 10 window selector menu items now have shortcuts
from "0" to "9".    Allows to easily switch channel windows with
Amiga-0..9.  - serious bug in DCC-SEND; neither RESUME-ACCEPT nor
MOVE-RESUME did work   since AmIRC didn't actually seeked to the
requested file position.  - added support for TIADCC (SetEnv TIADCC
"tiahost port"). The   variable format is compatible with Tau's DCC
implementation.  - fixed a few quirks in the string input class.  -
list class is now more compatible with fonts which have a "obscure"
tf_XSize value.  - sub processes weren't properly terminated during IPC
if the main application   was forcefully quit.  - fixed drag & drop
list sort for server and alias list (MUI3 only).  - the button/info row
in query windows can now be turned off to save   visual space.  - fixed
activation state of the user list. If the current user   PARTs or gets
KICKed, selection will turn off completely. If   the current user
renames himself, the selection will follow   accordingly.

0.65alpha quick fix :)
---------------------
- String input gadget was dead with MUI 2.3.  - added a kludge to avoid
MUI 3.0's MUIA_String_MaxLen-1-Bug within the   nickname gadgets

0.64alpha
--------
- fixed splitting and rejoining of lines in listview (spaces got
inserted   when a line couldn't be split on a word boundary) - main
listview now properly preserves text style attributes upon line
splits and relayouts - the ban popup list is now a "real" window which
can be dragged and sized - line read buffer pointer wasn't reset when a
server link was forcefully   terminated. Caused nonsense to appear on
next connect.  - added support for "DCC RESUME", a method of resuming
DCC SENDs   originally implemented by the mIRC for Windoze IRC client.
It's not   recommended to use DCC RESUME between AmIRCs, use MOVE
instead.  - OP/VOICE now accept up to four nicks - BAN did work with a
single nick only.  - the command parser assumed that all strings
beginning with "*"   are channel names, effectivly rendering /BAN
useless.  - KICKBAN window now has four buttons to setup "standard"
types   of ban mask (full, userhost, dynamic hostname, dynamic ip) -
CTRL-Cursor-Up/Down now allows to select the current user in   the user
list.  - TAB will now enter "/MSG nick " if the input line is empty
or just "nick" if there is already something entered in the line.  -
ALT-Cursor-Up/Down allows to scroll within the main output list.
ALT-SHIFT allows to page accordingly.  - fixed a few bugs in state
managment of the main list class.    Could cause enforcer hits when
```

```
LCM_INSERT was used before   MUIM_Show was called; this mainly occured
with the new   private message log which normally stays closed.  -
"Auto open Dialog Windows" was not functional at all.  - Function key
did only work when changed during the session, not   after loading.  -
Rexx script gathering caused entries with "PROGDIR:" to be   entered in
the list, which didn't work, cause it would   have referenced the Rexx
master process' PROGDIR: if   it would exist :) - several parts of
parsing code failed with characters > 127,   esspecially the notify
option with "strange" nicks.  - messages added to the private message
log now have a time   stamp.  - finally got around adding Troy Davis to
the thanks list   in the about requester.

0.63alpha
--------
- the vertical bar in the main listview was messed up.  - added VBars
on several of the other listviews (MUI3 only) - User list action
buttons didn't used MUIA_Font_Tiny as intended.  - "Save as" with no
config name given could cause erraneous behavior   and even crashes
with some requester patches.  - manually resetting AWAY via menu state
didn't restart the auto away   timeout.  - failure to open
"rexxsyslib.library" wasn't catched and could cause   serious trouble
later on.  - fixed a severe bug in the ioctl() call in AS225 which was
called   with the AmiTCP code for FIONREAD. Caused DCC's to loop dead.
- auto reconnect to server now limited to 10 retries.  - now checks for
duplicates before adding entries to the URL grab window.  - nickname
gadgets now accept all RFC-1459 allowed nickname characters   and even
"_" :) - added help bubbles for the channel mode buttons (MUI3 only)

0.62alpha
--------
- Click-On-Notify still didn't work <blush> - fixed selectwait() call
in the as225 tcp interface. DCCs are now   properly cancelable :) - /RX
did pass parameters, but only the first <blush again> - /NOTIFY did
react on parts of nicknames (e.g. "/NOTIFY OLI" reacted   on an
"OLIVER" being on) - added URL grabber; incoming text is scanned for
URLs and MessageIds,   which are presented in a list window. Double
clicking on a URL will   send them to AMOSAIC.1 via Rexx - Rexx: Using
"Say" with no open window caused terrible crash - added privmsg log
window which stores all private msgs send to you - added a few Undernet
servers to the default server list - added a few useful menu shortcuts

0.61alpha
--------
- incoming/outging DCC chat requests will now no longer open the dcc
chat   list window.  - startup errors (in particular wrong arguments
passed on a shell startup)   caused destructors to be inproperly
called, resulting in dead locks.  - WHOIS-ON-NOTIFY didn't work when
the watched user was logged in   (did send "WHOIS +USER"...) - made an
attempt to fix the as225 code which seems to be crashing   on DCC
(sorry, can't check this for myself) - /RX didn't pass arguments.  -
will now no longer create an empty Rexx menu if no rexx scripts are
found (AmIRC looks into PROGDIR:Rexx/ and Rexx: for files ending with
*.AMIRX).   It's nicer and avoids nasty crashes with certain buggy
Menu hacks :)

0.60alpha
--------
```

```
- channel list didn't worked at all.  - double click on an entry in the
notify list will now issue an /WHOIS nick   command.

0.59alpha
--------
- AWAY flag is cleared after connection to a server.  - Autojoin
channels are now bound to the server configs. This allows you to
specify different channels for different servers.  - state of the
"Create Icons" menuitem is now saved correctly.  - "Help" menuitem
works.  - Bug in the string class caused enforcer hits and made MUIKEYS
not   working correctly.  - when the connection was lost during receive
of the channel list,   the channel listview stayed deactived.  - empty
CTCP ACTION requests are no longer displayed.

0.58alpha
--------
- the light bulb's transparent color wasn't set correctly so the
background   got overwritten.  - fixed a few MUI object background
settings.  - program and basename now thoroughly changed from MICROIRC
to AMIRC.  - light bulb is now a page group to avoid MUIA_Bitmap_Bitmap
not being   correctly settable. Should also fix a potential crash.  -
added FKey support.  - String gadget is now constantly activated.  -
added "Invite" event.  - default prefs storage location is now
"PROGDIR:Default.MIRC" to   avoid problems if AmIRC is started from
another dir.  - now deiconifies upon receiving SIGBREAKF_CTRL_F

0.57alpha
--------
(skipped)

0.56alpha
--------
- light bulb/mic icon refresh now fixed.  - fixed QUERY windows to
correctly parse IRC formating codes on output.

0.55alpha (saturday evening release, please don't wonder :-)
------------------------------------------------------------
- fixed missing CTCP ACTION parsing for private PRIVMSGs - added
missing /DESCRIBE command.  - OP/VOICE status is now displayed with a
nice icon (thanks Morden).    The implementation of the display sucks,
it's gonna fixed in the next   release.

0.54alpha
--------
- fixed a bug in WALLOPS and ERROR parsing, not outputting anything but
 causing an enforcer hit instead.  - userhost database sometimes
caused calls to FreePool() with a mismatched   size (len-1). Could
screw up pool management severly.  - now entering a server name no
longer causes the list to be sorted,   leaving input focus on an
undesired entry.  - added missing /BAN, /BANDEL and /BANLIST commands.

0.53alpha
--------
- just bumped revision to avoid confusion with a "0.52alpha" floating
around.

0.52alpha (not published)
```

```
------------------------
- seperated user list now finally works without crashing badly.  - the
seperated user list windows now have a close gadget, and the channel
windows have a pop button to reopen it.  - fixed sorting of user list.
- changes of GUI settings now happen instantly. Note that MUI 2.x
requires closing and reopening of windows for some operations.  -
kludged around the userlist refresh bug by manually forcing redraw   on
every JOIN/PART.  - Auto-Rejoin-After-Kick now defaults to off due to
popular demand.

0.51alpha (not published)
------------------------
- fixed incorrect parsing of 317 extend message (signon time) - fixed a
few quirks in DCC SEND aborted output.

0.50alpha
--------
- fixed a crash with certain MUI versions on SAVE.  - DCC originating
sometimes didn't work since GetHostId() reported   "127.0.0.1"
(localhost). Now falls back to reading ENV:HOSTNAME   and doing
gethostbyname() on this.  - the user list may now be hidden totally or
put to a seperate window.    Note: This isn't implemented correctly
now; i'll change it ASAP.  - the user list sorting can now be turned
off.

0.49alpha
--------
- remote logging of AMIRC usage has been removed completely.

0.48alpha
--------
- added event handling. Allows you to deiconify, beep or play a
(datatypes) sound on certain events.  - some internal stuff.  - added
processing of the "250" extension message.

0.47alpha
--------
- fixed "DCC SEND/MOVE ABORTED" which was screwed in several places.

0.46alpha
--------
- added RAW IRC output window.  - Auto-Update-Get will now
automatically extract AMIRC.ReadMe to T:,   display it and offer to
automatically unpack the gotten archive   to RAM:
```

1.74 AmIRC.guide/Credits

```
Those Deserving Favour
**********************

Thanks to::
===========

The author wishes to thank the following people for their help during
```

```
AmIRC development:

Karl 'Phalanx' Bellve
     Who provides 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.

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 teh 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’ Hünerberg
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’ Gülicher
Danny ’Danny’ Amor
Markus ’Ill’ Illenseer
Markus ’Lammy’ Lamers
Matthias ’DrMabuse’ Lühr
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

1.75 AmIRC.guide/Bug Reports

How to Report Bugs
****

```
Bug Reports
===========


     Please send bug reports to AmIRC@lsd.wupper.de, 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@pgh.net
```

1.76 AmIRC.guide/Known Bugs

```
Problems, Causes, & Cures
*************************

Known Bugs
==========

   * A patch designed to eliminate requestors, easypatch,  will cause
     AmIRC .88 to crash.

   * 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.

   * Lockup when using Magic Menu. Install NewObtainGIRPort to cure
     this.  NewObtainGIRPort is available from aminet or from the AmIRC
     home page.  In older readme's, I suggested using SafeGIRPort, but
     there were reports that this won't really help the problem. Sorry,
     I don't use Cycle2Menu or MagicMenu and never had any problems
     with this myself.

   * The user list get's messed up. Known bug in MUI 2.3's listview's
     REMOVE method. Fixed in MUI 3.

   * 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.
```

1.77 AmIRC.guide/AmIRC Support

```
How to Get Help & Updates
*************************

One Click Updates
=================

You can easily download the latest version of AmIRC from the official
AmIRC support site by choosing the 'About...' Menu item from the
'AmIRC' menu. Here you will Find a gadget Labeled 'Download Latest
version of AmIRC'.  By clicking this gadget, AmIRC will check the
support site for the version of AmIRC there. If it is newer than the
version you are using, it will proceed to download it for you. If it is
the same version, you will be notified of this, and the download will
be aborted.  Once the new archive has been downloaded, You will be
shown the latest 'AmIRC.readme', and you will be given the option of
unarchiving the release to ram. NOTE: You must have LHA in your c:
directory for this functon to operate.  Once you have done this, just
overwrite the old AmIRC files with the new ones. Your configuraton
files will remain unchanged.

NOTE: The auto update will only download a stripped version of the
archives, because this assumes you have already installed the full
version of AmIRC.  with auto update, you will receive the executables
and any updated documents.

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@lsd.wupper.de
     with the text "Add" in the body.


     AmIRC FTP Support Site

     For AmIRC Updates & Patches, ftp to:

     ftp.vapor.com/support/amirc/

     AmIRC Support on the World Wide Web

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

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

1.78 AmIRC.guide/Index

Index
*****