xIrc User Reference

Joe Croft
croftj@dfw.net
http://www.dfw.net/ croftj

June 22, 1997

Contents

1 Introduction

xIrc is, as the name implies, an IRC Client program that runs under X. It is written in C++ and the binary for it provided was compiled using gcc 2.7.2 and qt-1.2 using the ELF libraries. See the INSTALL document provided for details on building it if you need or desire too, have no fear, it's reasonably easy to build.

I would appreciate any help offered in finding any bugs that may crop up. This will be mostly in the form of turning on the various debug flags I have embedded then mailing me the output. Also, any suggestions for additions or changes to it's look and feel will be appreciated. I cannot guarantee that they will be implemented, but all will be taken into consideration.

Also, if you know of servers not in my .servers file or any corrections for those that are, please mail me the changes for them and I will place them in the file for future releases.

Note the email address change. The old address is still there but...

1.1 Features

- Individual Windows for each channel you are joined to or each private conversation.
- Nick Lists for the channels that allow you to various things on the folks listed in them.
- Initiate and respond to DCC chat requests.
- Aliases that allow multiple words to be represented by a symbol. Similar to the IRC-II set command.
- Subset of similar commands and formats to IRC-II though many have been simplified.
- Auto responses to people getting kicked from a channel or for you being kicked from a channel.
- Auto re-join if you've been kicked from a channel.
- Simplified Banning w/ optional kick
- Use of X-Resources for the configuration of fonts, colors and a few other things.

- Cut and Paste functionality at least for the input fields few other things.
- Member arrival notification either by nick or by address mask
- Member ignore lists.

1.2 Future Dreams and Wishes

- Command line histories
- DCC File transfers
- Cut and Paste functionality for the chat windows in general.
- Script language based on the one provided by IRC-II.
- Anything else that you or I may think of that sounds handy, nifty or just plain fun to have.

1.3 Known Bugs

Still a bit flakey in the connect and disconnect portions of code but it has been getting better.

Known deficiancies in the QT-1.2 library make the edit portion of the Ports ComboBox in the xIrcConnectDialog unreadable. These should be fixed with the release od QT-1.3. The code *may* have to be recompiled and linked at that time to make use of the fixes.

Opens up a message query dialog for each private message if you don't have an open window open for the sender of a message when it comes in. I have found on some servers this can make for a very full screen when first connecting. Hopefully by the next release I'll have this fixed.

1.4 Credits

Troll Tech, the writers of the 'qt' class library used to make the X-Windows programming of this much easier.

The many folks who have given me ideas and inspirations for improving xIrc.

While I am at it, I would like to make an unsolicited plug for the folks who wrote the GUI class library I used, qt. This library was developed by the good folks at Troll Tech. Not only have they seen fit to provide the class library to us Linux folks for free, but they have also been invaluable in helping me making my program more bullet-proof. The library has a nice Motif like look and feel to it while it runs considerably lighter then Motif ever could and it greatly simplifies the writing of XWindows applications. They can be reached at info@troll.no and the URL to their home-page is http://www.troll.no/.

Any of you programmers out there might want to check it out if you are planning to do any X-Windows apps in the near future. Though it is not free fro most platforms, nor for use in commercial pursuits, it is free to the Linux community for non-commercial use. I have found it provides a nice interface the greatly simplifies the creation of windows and such.

2 Installation and Compilation

2.1 Installation

Installation is fairly straight forward. Choose whether you want to use the static linked version or the dynamically linked version. Copy whichever to a directory specified in your PATH.

If you are using the dynamically linked version, you need to install the qt library from Troll Tech. Please read their documentation closely so that you get the right version! If you are not recompiling my program you need the latest version compiled for gcc-2.72 (This is the 'standard' release). Otherwise you need to select the version for the version of compiler you are using. It sounds intimidating but if you read the docs supplied with qt carefully, it's not hard!

Next, copy the file xIrc.ppm to where you want it to reside. xIrc will look in the directories /usr/local/lib/xIrc, ./, and your home directory by default for this file. If you choose another home for it, you will have to place an entry, PixMapPath, in one of the resource files telling xIrc where too look. See the section X-Resources.

Move and rename the the file servers.dat to /usr/local/lib/xIrc/.servers. If you want to keep it elsewhere, The entry **Servers.Path* and **Servers.Filename* will need to be changed in the resources file.

If you are using fvwm window manager, I have supplied xIrc.xpm for the button's icon. I'll leave it to you to investigate setting up .fvwmrc to put all this to use.

2.2 Re-compilation

Change directories to the libX++ directory. Modify the CFLAGS entry to reflect the home of the qt include files. Then run *make depend* then *make*. This can be done on a single line as *make depend; make*.

After that completes, change to the xIrc-X.YY directory. Edit the entries QTINCDIR, QTLIBDIR, XPPINCDIR and XPPLIBDIR to point to their proper places. Then run *make depend* then *make*. This too can be done on a single line as *make depend; make*. After this completes, follow the installation notes for putting it and it's icon in their proper homes.

These files should compile w/ no errors or warnings. If they get either, I don't know what to tell you. I would suspect an older version of compiler or something. You can always email me and I'll help you as I can.

3 Usage

xIrc does not have any command line switches of it's own. The QT library may define afew of its own to change its look and feal, though I'm not sure what all are available.

3.1 Connecting to a Server

Upon starting, a dialog asking to choose your nick will be displayed as well as the main window. A default set of nicks are given. After selecting a nick, which you must do, a table of servers will be presented. Select your server of choice by double clicking the mouse on it. If your favorite server is not in the list, you may add it by selecting the *New* entry under the *Server* menu.

A small dialog will be displayed allowing you to change port numbers of manually type in a different server name. Once you are done, click on the *Connect* button.

At this point, xIrc will attempt to connect to the selected server. After it connects, any welcoming messages and the Message of the day will be displayed on the screen.

3.2 Entering a Channel

To enter into a channel, select the *Open Channel* entry in the *File* menu. You will be presented with a list of channels to enter and or people to chat with. Either select a name from the list or type it in in the edit field and press either *OK* or *Join* buttons. If you want to see who a person is or who is on a channel, press the Names/Whois button.

Once a channel is entered a new window will be dispayed for that channel. At this point any of the commands listed below may be entered or any non-command will be sent to the channel or person.

If a channel was entered, a list of the members will be displayed for the channel. If you find the list irritating as I do, you can press the *Nick List* button at the top of the channel window, it will toggle the display of the member list for the channel.

3.3 DCC Chat

3.3.1 Initiating a DCC Chat Connection

A DCC chat is initiated by either opening the *Channels & People* dialog by selecting the *Open Channel* entry of the *File* menu, selecting the person or entering in thier nick followed by pressing the *DCC Chat* button. Or by double clicking on the nick which will open the *Nick Action* dialog and pressing the *DCC Chat* button. In either case, a DCC Chat request will be sent to the nick, and a window will be opened for the conversation. When the other end connects, a small message will be displayed in the message window and you may be begin chatting with them.

3.3.2 Accepting a DCC Chat Request

When a DCC Chat request comes in, a message box will be displayed indicating who wants to chat with you. At this point you may either choose to connect to them or open a window to chat privately with them.

3.4 Ignoring People

Once in a while, jerks do enter in the scene and it is nice to be able to ignore them. This may be done in one of two ways. Either doble click on thier nick and when the *Nick Action* dialog is presented, press ignore. This will do a */who* command on them and build an appropriate adres filter for the person and present you with a dialog showing the results. Press the *OK* button to add the entry to the list.

The second means is to enter the name in manually. Select the *Ignore List* entry in the *Lists* menu of the main window. This will display the *Ignore List*. You may at this point double click on an empty row or select the *New* entry under the *Nick* menu of the table. This will open an dialog allowing you to enter in the nick and or address mask of the person.

Any message you enter in will be sent to the person each time xIrc receives a message from them. This includes messages from them to a channel you are in.

3.5 Notifications

The mechanics of setting up notifications of peoples comings ang goings are identical to that of the *Ignore* lists. The only difference is the use of the mesage field. It will be sent to the person whenever they first detected to be online. Kind of an auto-greets message.

3.5.1 By Nick or by Address

Both ignores and notifications allow you to select whether they are by nick or by address. Pressing the *By Addr* button indicates that it should be done by address. Otherwise, they are done by nick.

4 Windows, Dialogs and Buttons

There are several windows that may or may not be present at anytime through a session. Th following is a brief description of most of them.

4.1 Main Window

This window is primarily used for dealing with the server. If a message window, for either a channel or an individual, cannot be found for a message, it will be displayed in this window.

4.1.1 Menus

Enough actions have been added that I finaly had to replace the buttons with menus.

File Server and Channel related functions

> **Connect Server** Opens the Server Select Dialog (4.4) allowing you to connect to or disconnect from a server.
>
> **Open Channel** Opens a dialog (4.9) for joining a channel a channel or starting up a chat with an individual.
>
> **Quit** Opens up a dialog allowing you to select an exiting statement then exits the program

Nick Nick related functions

> **Show Folks Online** Toggles the visibility of the *Folks Online* list (4.11).
>
> **Actions** Opens the 'Nick Action' dialog (4.8).
>
> **Change Nick** Opens a dialog for changing your nick. If you select this, you will be forced to enter in a nick.

Lists List Access / Maintanence

> **Server List** Same as the Connect Server menu item under the File menu.(4.4)
>
> **Ignore List** Allow the editing and removing of existing, and the entry of new members on the ignore list (4.12.1).
>
> **Notify List** Allow the editing and removing of existing, and the entry of new members on the notification list (4.12.2).

4.2 Message Window

This Window is the window used for talking to individuals and channels. There can be many such windows open at any given time. The only real differences in the windows for chatting on a channel and those for chatting to individuals is their initial size when they are created. The windows for talking on a channel are created with about twice as many lines.

Also found in this window is an edit field for entering commands and messages to send. The commands and instructions on what can be entered within this field will be explained in later sections.

4.2.1 Buttons

Close Pressing this button parts (leaves) the the channel and closes the window. In the case of a window for speaking to an individual, the window is merely closed.

Ping Pressing this button send a CTCP Ping command to the individual or channel.

Nick Actions Pressing this button opens the same dialog box as the *Nick Actions* button of the Main window.

Nick List This button is only present on the Message Windows for channels. Pressing it either hides or shows a dialog box with the names of the members currently on the channel.

Clear Line Pressing this window erases any text currently present in the edit field.

4.3 DCC Chat Window

This window is for chatting to another person via the CTCP DCC Chat mechanism. It has the *Close* and the *Clear Line* buttons. These buttons have the same functionality as the same buttons on the *Message Window*

4.4 Server Selection Dialog

This Dialog Box is for establishing and dissolving connections with the Irc servers and manipulating the entries for the servers in the servers file.

The scrollbar, up/down arrow keys, page-up page-down keys, and the home & end keys can be used for traversing the list. Presing the Enter key is equivilent to pressing the *Connect* button unless a change has been made to the server selection masks.

The servers shown in the list can be limited by entering appropriate masks into the editable fields at the top of the table. These masks are used in a wildcard mode as in file selections. Pressing the Enter key after modifying any of these fields will update the table to display The only those entries that match the masks.

4.4.1 Buttons

This dialog has 3 buttons, *Connect*, *Disconnect*, and *Cancel*.

Connect This pressing this button opens the Connection Dialog which allows you to enter in a different server name or IP address and / or select the port inwhich to connect to.

Disconnect Pressing this button disconnects the server.

Cancel Pressing this button closes the dialog box without any action.

4.4.2 Menu Bar

The menu bar has two items, *File* and *Server*. The *File* menu has entries for manipulating the server file. The *Server* menu hes entries for manipulating individule entries in the table.

These are the entries in the *File* menu.

New This menu entry clears the table of all entries

Load This menu entry opens a file dialog to select and load a file of servers into the table.

Import This menu entry is like *Load* except that it imports server files in the format of mIrc's servers.ini files.

Save As This menu item saves the table to a file.

Done This menu item behaves as the *Cancel* button.

These are the entries in the *Server* menu.

Connect This menu entry behaves as the *Connect* button.

Disconnect This menu entry behaves as the *Disconnect* button.

Edit This menu entry opens a *Server Entry Edit* dialog bos for editing an enties contents.

Delete This menu item deletes an entry from the table.

4.4.3 Server File format

The file format for the servers file is quite simplistic by nature. It is a colon deleimited file with the following fields.

Group Name of the group of servers the server belongs to.

Country Name of the country or continent where the server is located.

State Name of the state or country where the server is located.

City Name of the city where the server is located.

Server Domain name or IP number of the server.

Ports Comma delimited list of ports that can be connected to.

An example of a valid entry is

```
BeyondIRC:US:MI:St.Louis:irc.primary.net:6660,6662,6664,6666,6668
```

4.5 Server Connection Dialog

This is a small dialog that gives the user an opportunity to change the server name and select the port to connect to the server before connecting to the server. This keeps one from being limited to only those servers and ports that are specified in the server list.

There are two editable fields, one of which is a combo-box, the server field and the port field.

There are also two buttons. The Connect button starts the connection, while the Cancel button closes the window without any further action.

4.6 Server Edit Dialog

This dialog allows for updating information in the Server Table.

4.7 Nick List Dialog

This dialog box is created and maintained for each *Message Window* opened for a channel. It holds the list of members of the channel and a series of buttons for various actions. If a nick is not present in the edit field, pressing a button will perform it's action on the currently highlighted entry in the list. Each button's action should be self evident by it's name but I'll explain them just in case.

4.7.1 Buttons

Who is Perform a */whois* command on the selected nick.

Chat Opens the *Channels / People* dialog box so a private chat or a DCC chat may be initiated with the selected member. The member's nick will be put in the edit field of the *Channels / People* dialog box.

Ping Send a CTCP Ping command to the selected nick.

Invite Invite the selected nick (Uh, only makes sense if you entered a non-members nick in the edit field) to the channel.

Kick Open the *Kick Message* dialog box to kick a member from the channel. This only works if you are a channel operator.

Clear Erases the contents of the edit field.

Close Should close the window but I don't think it does anything at this point. *Sheepish Grin*

4.8 Nick Actions Dialog

One of my favorite dialog boxes. This dialog box allows you to do all sorts of things. On the buttons that require a channel to be entered, you will receive an error if the channel field is blank.

Both the Nick and Channel fields will be filled in by double clicking on a word in a *Message Window* or the *Main Window*. This word selected will be placed in the nick field and the name of the window will be placed in the channel field.

4.8.1 Buttons

Dcc Chat Initiates a DCC Chat connection with the nick entered in the nick field and opens a corresponding *DCC Chat Window*.

Priv Chat Opens a *Message Window* for the nick entered in the nick field.

User Info Sends a CTCP UserInfo command to the nick entered.

Whois Sends a */whois* command for the nick entered.

WhoWas Sends a */whowas* command for the nick entered.

Finger Sends a CTCP Finger command to the nick entered.

Ban This command sends a */who* command on the nick entered, then builds an appropriate ban string and bans that person from the channel specified.

Kick Same as the *Ban* button except it kicks the person from the channel once they are banned. You will be given a dialog box to enter an appropriate message to go with the kick also. This button is always good for livening up a slow channel.

Time Sends a CTCP Time command to the nick entered.

Give Ops Performs a */mode +o* command for the nick and channel entered.

Take Ops Performs a */mode -o* command for the nick and channel entered.

Ping Sends a CTCP Ping command to the nick entered.

Who Sends a */who* command for the nick or list of names or whatever entered in the nick field. I found this to be the most useful for seeing in any of my buds are online by entering a list of domain name masks for each of them then pressing the button.

Notify Opens an edit dialog with a template *Nick* and *Address Mask* for inclusion in the *Notify* list. This command may respond a bit slow because it performs a */who* command on the nick to build its *Address Mask*.

Ignore Opens an edit dialog with a template *Nick* and *Address Mask* for inclusion in the *Ignore* list. This command may respond a bit slow because it performs a */who* command on the nick to build its *Address Mask*.

Close Closes the dialog box.

4.9 Channels / People Dialog

This dialog box is for joining channels and initiating either private or DCC chats with individuals. In Either case, an appropriate *Message Window* or *DCC Chat Window* will be created. As with the other dialog boxes with combo lists in them, pressing a button will perform it's action on either the name in the edit field, or the highlighted entry in the list.

4.9.1 Buttons

OK This button either joins a channel or opens a *Message Window* to the selected individual.

Join Does the same as the *OK* button.

DCC Chat Initiates a DCC Chat session with an individual and opens a *DCC Chat Window*.

Chat Does the same as the *OK* button.

Names/Whois Sends either a */names* or */whois* command to the selected channel or individual.

Clear Clears the edit field.

Close Closes the dialog box.

4.10 Nickname Dialog

This dialog lets you select or change your nick. As the others, it will either use the nick entered in the edit field, or the selected nick in the list. If the *Auto Nick Selection* button is enabled, the next nick found in the list will be used if the nick you attempt to change to is in use. At this time, if you bring up this dialog box, you **MUST** exit with the *OK* button.

4.11 Folks Online List

This is a small dialog showing the people from the *Notify List* (4.12.2) who are presently online. The nick that they are using and the address as specified in the list are displayed for each person.

Doulble clicking the mouse on an entry in the list will pop up the *Nick Action* dialog box (4.8) with the nick shown.

4.12 Ignore / Notify List Maintanence Dialogs

The list maintanence dialogs for both the Ignore and Notify Lists are quite similar in thier look and feal. Though a few names may differ, they a basicaly the same dialogs. They consist of a series of menus and a scrollable table with 4 columns.

4.12.1 Ignore List

The ignore list is for ignoring incoming messages from individuals. When a message from an individual on the ignore list is received, an optional auto-reply message may be sent back if one is present in the *Message* field for that person.

Currently, messages from ignored people are displayed in the main window. This will most likely change as I become more confident that it is working properly.

The only short comming of the auto-reply is that it will be sent to the person when they chat on a channel you are currently on. Of course, depending on your nature, this could be a good thing (snicker).

4.12.2 Notify List

The *Notify List* is to receive notification of a persons arrival or departure from Irc. Upon detecting a persons arrival, it will send an optional greating to that person and also post a message to the *Main Window* of thier arrival. When a person has been detected as having departed, another message will be posted to the *Main Window* staing so.

Detecting people is done with the *who* command using either thier nick or the name of thier provider. They both have potential problems. In the case of using thier nick, the problem is (at least to me) obvious, anyone can use a given nick so a notification may or may not be true. In the case of using thier hostname the problem is not so obvious. Doing a */who* on some hosts, there may be many people online from that host. *xIrc* gleens out the desired person by thier username, but if there are enough people on form a single hostname, you may get bumped because of the number of responses from the server. Yes, this is a real problem. if you want to see it happen, do a */who* with no name on a busy night. The list will be so long, chances are the server will knock you offline.

4.12.3 List Navigation and Manipulation

An entry is selected when its row appears to be sunken in respect to the other rows. Selecting a given row is a simple matter of using the up and down arrows or the page up and page down keys to select the next or previous item in the list. An item may also be clicked on with the mouse to select it.

As well as using the menus described below to mainipulate an item, the mouse may also be used. Double clicking on an item will cause an edit dialog to be opened for editing the contents. Double clicking on an unpopulated row will open a dialog for entering a new item.

4.12.4 Menus

File

- **New** Clears the list of all entries.
- **Load** Reads in an existing file appending it's items in to the list.
- **Save As** Saves the list to a file. The default filename given is that of the file that gets read on startup of the program.
- **Done** Closes the dialog.

Nick

- **Edit** Opens a dialog for editing the currently selected item in the list.
- **New** Opens a dialog with blank entries for appending a new item to the list.
- **Delete** Deletes the currently selected item from the list.

4.12.5 Fields

Each item in the list consists of 4 fields. Only the 3rd field differs between the notification and ignore lists and then only by it's label.

Nick Nick of the user to be ignored or notified of.

Mask Address Mask of the user to be ignored or notified of. The format of this field may be any string that makes for a valid /who query. In general, it will have a format of *nick@server*. because most people use dynamic IPs, the server will be prefixed with a *. As an example:

> peteh@*.satelnet.org

Ignore/Notify Checkboxes to enable / disable the item and to select if the user is to be detected by thier nick or by thier address mask.

Message For the Ignore List, if a message is present, it will be sent to the user automatically in response to any message detected by them. This include any messages they send to a channel you happen to be on also. (Yes, I can be quite nasty, snicker).

For the Notify List the message present in this field will be sent to the user on detection of thier arrival. It will be sent only once on each arrival of the user.

In either case, if this field of empty, no message will be sent.

5 Commands

Commands may be entered in the edit field of the *Message Windows*. They are distinguished from messages by the line starting with a '/' character. All white space at the beginning of the line is skipped, therefore, if you want to 'fake' a command, you must put a period or some other non-white space character before the '/', otherwise the command will be invoked! The following commands are available (Maybe others if I forget to add them to this list as I release the code).

ALIAS *tag string* This command allows you to have simple macro replacements. Though they are very basic, they can be useful. Basically *string* will be inserted anywhere $*tag* is found on an entered line. For example:

```
/alias hugs *Big Hairy Hugs*
/me gives so and so $hugs tightly
```

will result in an action of '**nick* gives so and so *Big Hairy Hug* tightly'.

AWAY *message* Sets you away. The *message* will be given to anyone who private messages you.

INVITE *nick* This invites a person to the channel associated with the window it is typed in on. It is functional but useless on windows used to chat with others privately

JOIN *channel* One of the commands you probably ought to leave alone. Use the 'Channel / People' Dialog instead. It will join you to *channel*, but the routing of messages will not right and therefore won't work!

KICK *nick* Kicks a *nick* from a channel. Once again, it's functional but useless if used from a *Message Window* not associated with a channel.

LINKS Lists the servers currently connected to your server. Don't ask me how to interpret it, I don't know!

LIST Lists channels topic and such. Works just like the LIST command in IRC-II.

MAP Similar to the **/LINKS** command except that it gives a more 'graphical' look to how the servers are connected. Though I still have to pretend I know what it is telling me.

ME Sends a CTCP ACTION message. In DCC Chat windows it will append the action to the characters `<=`.

MODE *modes* Sets the modes for the channel the window is associated with to *modes*. It works just like the MODE command in IRC-II except that the channel name is derived from the window's name.

NAMES [*channel*] Lists the members on a channel. If channel is not given, the channel name will be used. If I'm not mistaken, in the latter case, no results will be display, instead, they will be reflected in the Nick Dialog for that channel.

NOTICE [*nick*] *string* Sends a NOTICE to the Nick or channel specified.

NICK Changes your nick

PART Like the JOIN command, it there, but it's best to leave it be!

PING Pings the Member or Channel given. If neither are given, the name pinged is derived from the window's name.

PRIVMSG Sends a private message to the channel or member specified.

QUIT Closes the connection to the current server sending the optional message if present.

TOPIC Set or displays the topic for the channel associated with the window.

WHO Does a who on the nick provided. Works like IRC-II.

WHOIS Does a whois on the nick provided. Works like IRC-II.

WHOWAS Does a whowas on the nick provided. Works like IRC-II.

VERSION Sends a CTCP Version message to the channel or nick provided. If no name is given, the name of the window is used.

6 Command Line Entry and Edit Fields

Because both the *Message Windows* and the edit fields of the dialog boxes share a common widget for text entry, they have the same abilities when it comes to aliases and special character handling. Though it is not guaranteed that all of the special characters are appropriate or will even be acceptable in all of the dialog boxes.

6.1 Handling of Special Characters

Special characters may be entered using *escape sequences*. These are with the character \ followed by one or more characters. There are 5 flavors of escape sequences.

6.1.1 Escape Sequences

\b*string*\b This sequence will highlight *string*. String may be anything.

\u*string*\u This sequence will underline *string*.

number This will convert the decimal number *number* to a single character. Ei. \27 will send an ASCII *Esc* character.

\x*hex number* This will convert the hexadecimal number *hex number* to a single character. Ei. \1b will send an ASCII *Esc* character.

\0*octal number* This will convert the octal number *octal number* to a single character. Ei. \033 will send an ASCII *Esc* character.

6.2 Aliases

An alias may be used by entering `$<tag>`. Where tag is either a name previously defined using the */ALIAS* command or the name of an environment variable. Tag must begin with a letter, and may consist of letters, numbers and the underscore after that. The regular expression for aliases is `$[a-zA-Z_]+[a-zA-Z0-9_]*`. If an alias is found on the line but not defined it is replaced with "" (no quotes).

At this point, it is not possible to to immediately follow a $ with a letter such as $A without it being interpreted as an alias. The upside to this is that you may put in $456.34 without any special characters to pass on the $.

The names for alias are NOT case sensitive. $fred is equivilent to $Fred and $FRED.

6.3 Command Lines

A command line is considered any line entered in the edit field of either a *Message Window* or a *DCC Chat Window* that starts with a */command*. Any leading white space on the line before the / will be skipped. If the command is not recognized it is quietly ignored.

Any line that does not begin with a / is considered a message and is sent on to either the channel or the individual you are chatting with.

In either case, you may imbed aliases and escape sequences as needed.

7 Configurations Using the xIrc.defaults File

Currently there are two means used in the configuration of xIrc, the xIrc.defaults file and the X-Resources. I'm not sure if this will not change in the future or not. For Y'all who have been using xIrc for a while, you will find that somethings that used to be in the xIrc.defaults file have been moved and made into X-Resources. In the future, others or all may be moved also.

Any aliases that you want to pre-define, may be entered into this file. It will be read once on startup and then politely ignored through the remainder of the session.

At this point it MUST reside in the current directory where you are when you start xIrc. There are just a handful of various things you can set at this point, hopefully it will not get too much larger in time.

If you take a look at the defaults file provided, you will get an idea of how it's formated. definitions can be on multiple lines using the continuation character as the last character on the line before they new line. Any handy little definitions you want to be able to use while chatting can also be placed in this file and they will be there ready for you to use when you start up.

Here is the list of the currently used tags. Not all of them are needed, though some do make life much easier. These names may be in any case to simplifiy reading them.

BAN_MESSAGE String to use for the kick if the kick is implemented with a ban.

CHANNELS List of channels or nicks you want to regularly check out, chat with or whatever.

DCC_SEND_RESPONSE Response message you want to send when someone tries to set up a DCC File transfer to you.

EMAIL_ADDR Entry to specifiy the user name and host name provided when connecting to the server. It must follow the ¡user¿@¡domain¿ syntax or you stand the chance that your connection will be rejected. If this entry is not provided, the user name from the /etc/passwd file will be used and a dummy domainname of dummy.hostname.org will be used.

KICKMESSAGE Default message to send when you kick someone.

KICKED_OTHER_RESPONSE Message to send when a kick of someone is detected.

KICKED_YOU_RESPONSE Message to send someone when they kick you.

NICK Now Obsolete tag. See NICKS.

NICKS Allows you to define a set of possible nicks to when connecting. Each nick should be seperated with a space.

QUITMESSAGE Default message to append to the QUIT command.

REALNAME Will be sent when first connecting to a server as your real name. Can be anything you like though if you're one of those snots that don't like people being able to see just who they are really talking too (Personal oppinion there). If it is not defined, the user_name field of the /etc/passwd file will be used instead.

SERVERS List of server names you frequent.

TCPPORT Default TCP port, probably should be left at 6667.

USERINFO Whatever you want to tell the world about yourself when they do a ctcp USERINFO command of you.

7.1 Obsolete Tags

These tags are no longer used as of Version 1.15 and their functionality has been moved to the X-Resources file.

FRAME_FG_COLOR_NAME

FRAME_FG_COLOR_RGB

FRAME_BG_COLOR_NAME

FRAME_BG_COLOR_RGB

WINDOW_FG_COLOR_NAME

WINDOW_FG_COLOR_RGB

WINDOW_BG_COLOR_NAME

WINDOW_BG_COLOR_RGB

8 X-Resources

As of version 1.15, xIrc uses X-Resources to set the colors and fonts to be used when drawing the widgets. Almost all of its widgets have both a class name and a resource name. Currently the names are the same except that the class names have the first letter of each word capitalized. The names and definitions of the font and color resources come directly from the definition of the Qt class library. Therefore, for a better understanding of them, it wouldn't hurt to take a look at Qt documentation for the classes QColorGroup and QFont.

Currently xIrc does not use the resource definitions defined for the root window. They are derived by merging the following sources in the order given.

- Hardcoded program defaults
- /usr/lib/X11/app-defaults/xIrc*
- /.Xdefaults

Note:

> * This file name will be the same as the the name that xIrc was invoked as. In otherwords, if a link name **fred** points to the actuall xIrc program, the resulting file name used will be /usr/lib/x11/app-defaults/fred.

The result of this is that if no other source is given, then the hard coded valuee will be used, otherwise, and values given in the .Xdefaults will overload those in the app-defaults file which will overload the hardcoded values.
To simplify the documenting of the resources and the widgets, I will refer to all of them by thier class name. If you insist on using the actual names, you can convert them by making the class name all lowercase.

Note:

> For now, it is recomended that only class names be used in the resource files. This is because in the near future the names for each widget will probably be changing and I'm not sure how yet. Where as I see very little change (though anything is possible) of the class names.

8.1 Default Values

The xIrc program has the following defaults built in. They will be in force unless over ridden by definitions in one of the aforementioned files.

```
XIRC*PixMap: xIrc.ppm
XIRC*PixMapPath: ./;~/;/usr/local/lib;
XIRC*Font.Family: Helvetica
XIRC*Font.Size: 12
XIRC*Font.Weight: normal
XIRC*Menu.Font.Weight: Bold
XIRC*Popup.Font.Weight: Bold
XIRC*PushBtn.Font.Weight: Bold
XIRC*PushBtn.Background: #bfafaf
XIRC*Background: 0xc3c3c3
XIRC*Forground: black
XIRC*BaseColor: white
XIRC*TextColor: black
XIRC*AutoFocus: false
XIRC*ListBox.BaseColor: #af9f9f
XIRC*ListBox.Background: #af9f9f
XIRC*MessageDialog.Background: #bfafaf
XIRC*MessageDialog.PushBtn.Background: #af9f9f
XIRC*MessageDialog.MultiLineEdit.Font.Family: Fixed
XIRC*MessageDialog.MultiLineEdit.BaseColor: #af9f9f
XIRC*MessageDialog.MultiLineEdit.TextColor: Navy Blue
XIRC*DccChat.Input.Font.Family: Fixed
XIRC*MsgChat.Input.Font.Family: Fixed
XIRC*ServerDialog.Font.Family: Fixed
XIRC*ServerDialog.ImportFile: servers.ini
XIRC*ServerDialog.ImportPath: ./
XIRC*ServerDialog.ImportFilter: *.ini
```

```
XIRC*ServerDialog.File: .servers
XIRC*ServerDialog.Path: /usr/local/lib/xIrc
XIRC*ServerDialog.Filter: .*
```

I have tried to keep the defaults to a minimum and as generic as possible allowing for the greatest flexiblilty. If they cause any problems, I'm open for suggestions on how they should be set, though I don't make any promises everybody's suggestions wil be put into place.

8.2 Common Resources

Each widget has these resources named.

Background Used to se the backgound color of the widget. On many of the widgets, such as lists, this also includes any scroll bars.

Foreground Not sure what all the covers. The one thing that I have noticed that it changes is the caret color in the editible fields.

BaseColor Background color of the input fields. The Background color will be used if this is not set.

TextColor Color of the text that is displayed in a widget. The Foreground color will be used if this is not defined.

Font.Family Family name of the font to use, such as, Helvetica, Courier, Fixed, etc.

Font.Size Pixel size of the font to use.

Font.Weight Weight of font to use. This can be either a number from 1 to 99 or one of the following names.

- Light
- Normal
- Bold
- Black

8.3 Configurable Widgets

XIRC The application. This has following resource names in addition to the common ones.

- **BanDialog** The dialog box used for banning folks from a channel. It has the **Input** and **PushBtnFrame** widgets beneath it.
- **ChannelDialog** The dialog box for selecting the channel to talk on. It also has the **ComboBox** and **PushBtnFrame** widgets beneath it.
- **ChannelNickDialog** This dialog box holds the list of current members of a channel and has buttons for various actions for the nicks. It has the **ComboBox** and **PushBtn** widgets beneath it.
- **ComboBox** This is for widgets composed of a list box and an edit field. It has the **Input** widget beneath it.
- **DccChat** Window used for DCC Chats. This has the same widgets beneath it as the **MsgChat** widget.
- **ErrorDialog** The dialog box that pops up when an error is recieved from the server.
- **IgnoreDialog** The Dialog Box for viewing and altering the *Ignore List*. It has the following resources in addition to the standard resources.
  - **File** Default name for loading a file into the list.
  - **Filename** Name of the file to load on startup.
  - **Filter** Filter for the files to see in the file dialog.
- **Input** This widget is a single line editable field.
- **ListBox** Used to set the colors and fonts of the various list boxes.
- **Main** Main text window that the server messages go to.
- **MessageDialog** The dialog box that pops up when someone sends a private message to you. It has the **PushBtnFrame** widget beneath it.
- **MsgChat** The window you talk with to both a channel or another person privately. Currently ther is no differentiation between the two, though later there will

probably be so. I will keep the ends of the new class names **MsgChat** to simplify the modifications to the resources when that time comes.

This has the **MultiLine**, **PushBtnFrame**, **ChanNickDialog**, and **Input** widgets beneath it.

MultiLine.Frame This specifies the color of the frame and scroll bar sourounding the multiline text widget used for displaying the text of the conversations. At this time, because of how the frame is layed out, it really only effects the color of the scrollbar.

MultiLine.Window The multiline text widget used to display the text of conversations. For the best results, a fixed font such as lucidatypewriter or fixed should be used for this window. The code for the widget was written on the assumption that this would be so, therefore, all bets are off that the widget will display it's contents correctly if a proportional font is used!

In addition to the standard fonts and color resources that may be configured, it also include the resource **HighColor** for configuring the color of any bolded text.

NickDialog The dialog box for selecting a nickname. It too has both the **ComboBox** and **PushBtnFrame** widgets beneath it.

NickActionDialog My favorite, the dialog box to do just about anything to anyone. It have the **Input** and **PushBtnFrame** widgets beneath it.

NotifyDialog The Dialog Box for viewing and altering the *Notify List*. It has the following resources in addition to the standard resources.

Menu Used to set the colors and fonts of the menu bar.

Popup Used to set the colors and fonts of the popup menus.

File Default name for loading a file into the list.

Filename Name of the file to load on startup.

Filter Filter for the files to see in the file dialog.

PixMap Name of the Pixmap file to use as an Icon for closed *Message Windows* and *DCC Chat Windows*.

Unfortunately, This file cannot be of the standard xpm variety. It must be a BMP, XBM, or a PNM of the type P1 - P6. I have supplied both an XPM and a P3 format PPM file derived from the XPM. The XPM version is for all of you fvwm users.

PixMapPath A list of paths, delimited with semicolons, to search for the pixmap. The default is './; /;/usr/local/lib'.

PushBtnFrame This frame is for holding one or more push buttons. It has the **PushBtn** widget beneath it.

ServerDialog The dialog box for selecting a server to connect to. It has the **ComboBox**, **PushBtnFrame** and **Input** (for the port number) widgets beneath it. In addition to the common resources it also uses these additional resources.

Menu Used to set the colors and fonts of the menu bar.

Popup Used to set the colors and fonts of the popup menus.

ImportFile Default name for the mIrc servers.ini file

ImportPath Default path to find the mIrc file.

ImportFilter Default filter for the mIrc file.

File Default name for the server file

Path Default path to find the server file.

Filter Default filter for the server file.

SocketDialog The dialog box showing the progress of the connection attempt. It has the **PushBtnFrame** widgets beneath it.

TCPSocket The socket widgit for establishing and accepting connections. This is truely not a widgit as it is never displayed on the screen. This object has the following two configurable items.

ConnectTime Time to wait to make a connection in milliseconds (seconds * 1000). The defualt is 1 minute.

AcceptTime Time to wait for an incomming connection in milliseconds (seconds * 1000). The default is 5 minutes.