

The KSirc Handbook

**Andrew Stanley-Jones, Philip Rodrigues, and
Anne-Marie Mahfouf**



The KSirc Handbook

Contents

1	Introduction	1
2	Using KSirc	2
2.1	If you haven't used IRC before...	2
2.2	KSirc Basics	2
2.2.1	Servers	2
2.2.2	Channels	3
2.2.3	Channel Options	3
2.2.4	User Options	3
2.3	Autoconnecting	4
3	KSirc Configuration	5
3.1	Introduction to Configuration	5
3.2	The Look and Feel Tab	5
3.2.1	Window Mode	5
3.2.2	Wallpaper	5
3.3	The General Tab	6
3.3.1	Global Options	6
3.3.2	Per Channel Options	7
3.4	The Startup Tab	7
3.4.1	Server	8
3.4.2	Name Settings	8
3.4.3	Notify List	8
3.5	The Colors Tab	8
3.5.1	Chat colors Configuration	8
3.5.2	Nick Colors	8

The KSirc Handbook

3.5.3	Color codes	8
3.6	The User Menu Tab	9
3.7	The Server/Channel Tab	9
3.7.1	Server	9
3.7.2	Channels	9
3.8	The Auto Connect Tab	9
3.8.1	Auto Connect Setup	9
3.9	The Fonts Tab	10
4	KSirc Colors	11
4.1	mIrc compatibility	11
4.2	KSirc native	11
4.3	Color Numbers	12
4.4	Sending Bold, Underline, Reverse, and Color	12
4.5	Examples	13
5	Filters	14
5.1	Filter Rules and How to Make them	14
5.2	Examples:	15
6	Keys	18
7	Tips	20
7.1	Joining KDE channels	20
7.2	Auto identify	21
8	Credits and Licenses	22
A	Installation	23
A.1	Where to get KSirc	23
A.2	Compilation and Installation	23

List of Tables

6.2	Keybindings	19
7.2	KDE IRC channels	20

Abstract

This documentation describes KSirc 1.2.1, the KDE IRC client.

Chapter 1

Introduction

KSirc is the default KDE IRC client. It supports scripting with Perl and has a lot of compatibility with mIRC for general use.

Chapter 2

Using KSirc

2.1 If you haven't used IRC before...

Simply put, IRC is a chat protocol, defined by official Internet standards, and capable of stealing many hours of your life. To use IRC, you must connect to a server, and then join a channel (equivalent to a chat room).

To connect to a server, open KSirc and select Connections → New Server... or press **F2**. In the dialog box that is displayed, select a group from the leftmost drop-down box and then a server from the middle drop-down box. The port, shown on the right, will almost always be 6667. A new window will be displayed showing the messages sent by the server to the client. When these messages have finished scrolling past, you want to choose your channel. To display a list of available channels, type **/list**. To join a channel, type: **/join #channelname**. To leave a channel, just type **/part**.

That's the basics; for more detailed information, read on...

2.2 KSirc Basics

2.2.1 Servers

KSirc allows you to connect to any number of servers. To connect to a server, select Connections → New Server or press **F2**. The Connect to Server Dialog appears, and you can select a Group, then a Server, and also a Port to connect to.

Just click Connect when you're done, and a new window for the server is displayed. You can keep track of the servers you're connected to in the KSirc Server Control window.

You can get help on most IRC commands by typing `/help commandname`. Typing `/help` on its own gives a list of available commands.

2.2.2 Channels

You can connect to channels with the normal IRC command `/join #channelname`, by selecting Connections → Join Channel... or by pressing **F3**.

Channels you are connected to are shown in the Server Control window under the server to which they belong. To leave a channel, you can `/part`, select Channel → Close or press **Ctrl+W**.

2.2.3 Channel Options

The Channel menu, available in any channel window, contains some useful options:

Save to Logfile... (Ctrl-S) Saves the contents of the channel to a file.

Time Stamp If selected, prepends each thing said in the channel with the time it was said, in the form [HH:MM:SS]

Ascii Table (Alt C T) Displays a table of Ascii characters, from which you can choose any characters you require.

Beep on Change (Alt C P) If selected, KSirc emits a beep every time something is said on a channel.

2.2.4 User Options

In a channel, the user list is displayed along the right hand side. Right-clicking a nickname in that list gives you a pop-up menu of options, most of which are self-explanatory. Those actions can be configured (changing the order, adding or removing some) in the Configure KSirc... dialog in the User Menu tab. The default actions are the following:

Refresh Nicks Sends a message to the server to request the list of users in the channel. This ensures that the list of nicknames is up-to-date.

Follow Highlights the user's nick in every message they send to the channel. This way, you can follow someone's thread of conversation in a busy channel. The user's nick color is picked at random.

unFollow Stops following the user.

Whois Sends a CTCP whois query to the user, which provides information about the user, such as his hostname and what server he is connected to.

Ping Sends a CTCP ping request to the user, to determine his echo time.

Version Sends a CTCP version request to the user, to determine what software he is using.

The KSirc Handbook

Abuse Abuses the user in one of various amusing ways.

The following commands are only available if you are an operator on the channel:

Kick Removes the user off the channel.

Ban Bans the user from joining the channel.

unBan Removes the ban on the user joining the channel.

Op Gives the user operator privileges.

DeOp Removes operator privileges from the user.

2.3 Autoconnecting

If you have a number of servers and/or channels which you connect to every time you use KSirc, then you can set up KSirc to automatically connect to them: In the Server Control window, select Settings → Configure KSirc..., and in the dialog which appears, select the Auto Connect page. To add a channel on a new server, fill in the Server and Channel text boxes, and then click on Add. To add another channel on this server, select the server name in the tree view, enter the channel name in the Channel text box, and click on Update/Add. You can repeat this process as many times as you like to add several servers and channels for KSirc to connect to at startup.

Chapter 3

KSirc Configuration

3.1 Introduction to Configuration

The KSirc configuration dialog is available from Options → Preferences... in the KSirc Server Control window.

3.2 The Look and Feel Tab

3.2.1 Window Mode

SDI Mode (old behavior) Causes KSirc to use single document interface mode, in which each new channel or **/query** window has its own, separate window.

Paged MDI mode (XChat) Causes KSirc to use multiple document interface mode, in the style of XChat, in which each new channel or **/query** window has a tab in one main KSirc window. Each tab can be brought to the front by clicking its tab, and you can switch tabs using Alt-Left Arrow and Alt-Right Arrow.

3.2.2 Wallpaper

In the Wallpaper section, you can change the image placed in the background of your KSirc windows. You can enter the name of the file to use in the text box, or browse for files using the button to the right of the textbox. Image files used must be in GIF, JPEG or PNG format. A preview of the image is shown at the right of the dialog box.

If you don't want a wallpaper, simply right-click in the path field and choose clear in the context menu. Or remove the wallpaper location to leave the field empty and click Apply. No wall paper will be set then.

3.3 The General Tab

3.3.1 Global Options

Dock in system tray This allows KSirc to be docked in the system tray. By default this is not enabled. When KSirc is docked in the system tray, you are able to access several options by clicking on the KSirc icon. When you close KSirc window, the icon stays in the systray until you quit KSirc.

Color picker popup If selected, a popup window from which to select the color of your text is presented when you press Ctrl+K. If not, you have to type the color codes manually. See also chapter 4. This option is selected by default.

Auto create window If selected, KSirc will automatically create a new window for each user who sends a `/msg` command to you. If not selected, any text sent to you with `/msg` is displayed in the current window and you can use `/query username` to create a window to chat to that user. It is selected by default.

Auto-rejoin Rejoins channels automatically if you are disconnected.

Auto create on notice If someone sends you a `/notice` then if this option is checked it will create a new window.

Announce away messages If this is checked, you will see the messages when a user selects the away option. By default this option is not checked. It is checked by default.

Nick completion If selected, switches nickname completion on. Nickname completion works as follows:

1. Type the first letters of a user's nickname.
2. Press TAB. The text you typed will be completed to match the username, including changes in capitalization if necessary. e.g. `phi<TAB>` becomes `PhilRod` if there is a user called 'PhilRod' on the channel.
3. If more than one user's nickname on the channel matches the text you have typed, the first name in the list is chosen. Pressing TAB subsequent times displays the next nickname in the list. For example, if there are users 'PhilA' and 'PhilRod' on a channel, and you type `phi<TAB>`, the text will first be completed to `PhilA`, and if you press TAB again, it will change to `PhilRod`.

Use color nick list If selected, it will use the colors set in the Colors tab of the Configure KSirc dialog for coloring the nicknames.

Display topic in caption Displays the topic of the current channel in the window caption. If not selected, the topic is only displayed inside the window.

Dock passive popups This option is only available if Dock in system tray is checked. If it checked, the passive popups will be displayed docked to the system tray.

One line text entry box If this is not selected, the entry box where you write your text in each channel will expand on several lines depending of the text length. If this is selected, only one line will be used and that means you will not see the beginning of the text you are writing if it is longer than the line.

History Length Stores up to this many lines of chat from each window, allowing you to scroll upwards and see what has already been said.

Auto save history If this is selected, private message history is saved. When you open a private window the second time with the same user, you will see a log (the history) of your previous chat with him.

3.3.2 Per Channel Options

Override existing channels options If this is selected, the settings in this tab will override each channel options so these settings will be applied in each channel, independently of your channel settings in the Channel menu. This setting will only work until next time you open the configuration dialog and it will be reset unchecked then. This is because you probably don't want to override the existing channels options all the time.

Time Stamp Prepends each thing said in the channel with the time it was said, in the form [HH:MM:SS].

Show topic Displays the channel topic on top of each channel window.

Beep on change If selected, a system beep will be generated whenever a user on your `/notify` list signs onto the IRC server you are on.

Enable logging Creates log files for each channel per day. These log files are stored in `$(KDEHOME)/share/apps/ksirc/logs/` and the files look like: `$(KDEHOME)/share/apps/ksirc/logs/#channelname_year_month_day_servername.log`

Hide part/join messages Don't show part/join messages. This is useful in a channel with many people.

Default encoding Fixes the default encoding for all channels. Choose the encoding in the drop-down list.

3.4 The Startup Tab

Here you can configure KSirc's startup settings.

3.4.1 Server

3.4.2 Name Settings

Nick Name Set your IRC nickname.

Alternative Nick Set the nickname to use if your first choice is already in use by another user.

Real Name Your real name, as returned by a **/whois** query on you.

User ID Your real name, as returned by a **/whois** query on you.

3.4.3 Notify List

Here you can enter a list of users who you want added to your **/notify** list at startup. You will be alerted when each of the users in the list is online.

3.5 The Colors Tab

Here you can configure KSirc's use of color.

3.5.1 Chat colors Configuration

Here you can set colors for each of the types of text. Clicking the color button to the right of each text type's label will display a color selector box in which you can choose the color in which you want this type of text to be displayed.

3.5.2 Nick Colors

Foreground: and **Background:** set the colors for the foreground and background of users' nicknames in chat windows. The **Color for messages containing your nick:** option sets the color for any message appearing in a channel which contains your nickname.

3.5.3 Color codes

Here you can set whether KSirc allows native color codes (see chapter 4) and also whether mIRC color codes are allowed.

3.6 The User Menu Tab

This page allows configuration for the RMB (Right Mouse Button) menu for the nick list located on the right in each channel. When you right click on a nickname, some actions are defined by default. These are the actions listed on the left. You can change these actions order, delete some and add others.

3.7 The Server/Channel Tab

Here are listed the servers and channels you join to using the New Server (F2) and New Channel (Ctrl+N) dialogs via the Channel menu.

3.7.1 Server

You add there the servers you want to use. Write the server name in the field and click the Add Server to List button. If you want to delete a server from the list, you click on the server name and then on the Delete Server from List button.

3.7.2 Channels

You add there the channels you want to connect to. Write the channel name in the field and click the Add Channel to List button. If you want to delete a channel from the list, you click on the channel name and then on the Delete Channel from List button.

3.8 The Auto Connect Tab

3.8.1 Auto Connect Setup

You set here the server and channels names that you want to connect directly when KSirc starts.

You need to use the Auto Connect Setup dialog first to add new channels. You enter there the server name. The server port is set by default for most servers. If you are not sure, just leave it as it is. Usually servers don't ask for passwords so if you don't have a password, leave the Password field empty. Then add a channel name and click the Add button. The channel will be displayed in the Auto Connect List.

If you want to add more channel for a same server, click on the server in the Auto Connect List and the server name will be displayed in the Auto Connect Setup dialog. Write the channel name in the Channel field and click on Update/Add.

In order to remove a channel form the Auto Connect List, click on the channel name in the list and then click the Delete button.

3.9 The Fonts Tab

You can change here the font for the main window. Select the font you want then click the Apply button to preview the change. Ok applies the changes and quits the dialog.

Chapter 4

KSirc Colors

KSirc follows the color scheme used by mIrc and a slight modification for more powerful in house use.

```
<fg> == foreground  
<bg> == background  
[] == optional
```

4.1 mIrc compatibility

Format:

- `0x03<fg>[, <bg>]` sets the foreground and background
- `0x03` resets to defaults for *that* line

4.2 KSirc native

Format:

- `~<fg>[, <bg>]` sets the foreground and background
- `~c` resets to defaults
- `~b` sets bold font
- `~u` sets underline
- `~i` sets italics
- `~r` sets reverse video

Why did I change KSirc to use `~` instead of `0x03` (Ctrl+C)? Well, it's hard to use `0x03` in scripts and not all C functions seem to like it. `~letter` also allows more commands while not stomping on mIrc's future changes.

4.3 Color Numbers

1. white
2. black
3. blue
4. green
5. red
6. brown
7. purple
8. orange
9. yellow
10. lightGreen
11. cyan
12. lightCyan
13. lightBlue
14. pink
15. gray
16. lightGray

4.4 Sending Bold, Underline, Reverse, and Color

You can use the following key combinations to insert control codes in text:

- Ctrl+B for bold text
- Ctrl+U for underlined text
- Ctrl+R for reverse text
- Ctrl+K for colored text
- Ctrl+O for plain text

4.5 Examples

To underline a single word in a sentence:

1. Type Ctrl+U
2. Type in the word
3. Type Ctrl+U again

Only the text that is enclosed by the start and end codes will be affected. You can use this method with all of the other control codes.

The Ctrl+K control code is slightly different because it allows you to specify a color number. To color a single word in a sentence:

1. Type Ctrl+K
2. Type a number between 0 and 15
3. Type the word
4. Type Ctrl+K again

If you also want to change the background color of a word, you would need to type two numbers separated by a comma instead of just one number. The first number is the text color, the second number is the background color. The colors range from 0 to 15, the index is in the previous section.

You can enclose text in multiple control codes, so for example you could have a bold, underlined, and colored word.

Chapter 5

Filters

5.1 Filter Rules and How to Make them

If you just can't figure it out, wait. I want to build a nice 'filter builder' where you can just click your way through it. Though, it might be a while.

The filter tries to find the 'match' string then use the 'From' and 'To' as a substitution. The match, from and to are all Perl regex expressions. Rules are evaluated in descending order. The top rule is evaluated first, then the second from the top, etc. All strings are evaluated as:

`$$name` is expanded to the environment variable `name`. This is done immediately when you insert the rule, and will not change after that time. Therefore it's probably of limited value.

`$$name` is substituted with the Perl `$name` variable during the match. This can be substrings such as `$1`, `$2` in the substitution, or normal variables available under sirc (such as `$nick`, `$channel`, etc.).

`˜name~` prepended *once and only once* to the line will send the line of text to the window called `name`. If the window does not exist it will go to the last window which had focus. There are several special windows, all prefixed by a single `!`:

!default The current default window. Guaranteed to exist.

!all Send to every window. This might not show up on all windows, depending on how the text is parsed. For example, channel windows won't show a `/part` or `/quit` unless the nickname is on the channel.

!discard discards the text.

The rest of the expression is dealt with as normal Perl regex. A good understanding of the `perlre` man page will definitely help, but a basic understanding of regex is most certainly required.

5.2 Examples:

1. Want to convert all boren from boren to BoreN

```
Match: .*
From: boren
To: BoreN
```

Pretty straight forward, match anything, then substitute boren with BoreN.

2. You want to match everything with boren in it and send to the window called 'boren'

```
Match: boren
From: ^
To: &#732;boren&#732;
```

Looks for 'boren' if found, substitutes the beginning of the string (^) with ~boren~.

3. Though the previous example works, if the string already has ˜so-mewindow˜ on it, you'll now have two ˜boren˜˜so-mewindow˜... So you can do this instead.

```
Match: boren
From: ^(?:&#732;\S+&#732;)
To: &#732;boren&#732;
```

Ok, the from line is a little bit more complicated. It means: match 0 or 1 copies of ˜\S+˜. Which is 1 tilde, one or more non-whitespaces, and then another tilde. The paranoid might do `(*:˜\S+˜)`, which means: match 0 or more channel directives in case prior rules are broken.

4. Server kill messages tend to be long, ugly, annoying, etc. A basic message on dalnet looks like:

```
*** Notice -- Received KILL message for
BOB0!ANDY@line82-basel.datacomm.ch from NickServ
Path: empire.ny.us.dal.net [209.51.168.14]!trapdoor.ca.us ←
      .dal.net
      [206.86.127.252]!caris.ca.us.dal.net [208.1.222.221]
      !services.dal.net [2008.1.222.222]!services.dal.net
(NickServ Enforcement)
```

When you're +s you get tons of them; you don't want all of them flying across your screen. I'm going to make 3 rules to deal with them one bit at a time. You could do it in less rules, but it'll show you the basic rule structure, in nice steps, and how to use multiple rules to parse a message. The first step is to remove the Path: portion of the message:

The KSirc Handbook

```
Match: ^*\*\*\. * KILL message for.*
From: Path: \S+
To: .
```

Match looks for the message starting with `***`, the `*`'s have to be quoted with `\` since by themselves they mean 0 or more of the prior character. `.*` means: match anything until you find `KILL message for`. This allows us to avoid typing in `-- Received...` etc. The trailing `.*` means: match anything to the end of the line. (not needed, I think)

The From line means: substitute " Path: " and any non-whitespace characters with the To. To is a "." therefore the entire path turns into a single period.

The message now looks like:

```
*** Notice -- Received KILL message for BOBO!ANDY@line82 ←
    -basel.datacomm.ch
from NickServ. (NickServ Enforcement)
```

Notice the new "." after NickServ?

5. Ok, the message is a lot cleaner, but KILLS from nickserv aren't really that important, so let's forward them to the !discard window.

```
Match: ^*\*\*\. *KILL message.*\ (NickServ Enforcement\ )
From: ^(?:&#732;\S+&#732; )
To: &#732;!discard&#732;
```

Match rule searches for the KILL message and makes sure it's from NickServ. Notice the `\ (` and `\)` are both used in regex, therefore we have to quote them. This is very similar to what we said two examples before.

6. We've now filtered out all the nickserv kills, but the message is still pretty hard to read by simply glancing at it. So let's reorder it to something like:

```
*** [KILL] KILLER; killed KILLED; (REASON)
```

```
Match: \*\*\*\. *KILL message
From: \*\*\*\. *for (.*) from (.*)\ . \((.*)\) .*
To: *** [KILL] $$2 killed $$1 ($$3)
```

Ok, the match looks for `***something KILL message`. We can't use `^` since we may have just appended `˜<window>˜`.

The from line gets a little more interesting. The "for (.*)" looks for the word "for" then some text. `.*` means: match zero or more of anything except newline, but isn't greedy. The rule is to stop when the first terminating condition is found, not the last. In other words it matches anything until a space is encountered. The surrounding `()` means: save the contents. Each `()` saves the matched data in `$$#` where `#` starts at 1 for

The KSirc Handbook

the first substring, etc. In this case, \$1 gets the nick/user-info of the person killed. \$2 is then filled with the name of the killer. Between the () we have the reason for the kill. Here the (and \ (get a little confusing. Remember \ (matches the actual character ' (' .

7. How to colorize your life.

Ok, you want to add some color to KSirc. See the [Colors section](#) for color info, but here's a filter rule to highlight the nickname between <NICK> on each line:

```
Match: ^(?:\S+\S+)<\S+>
From: <(\S+)>
To: <\S+\S+\S+>
```

Takes the nickname and adds color #4 between the two <>. \S+;c clears the color.

Chapter 6

Keys

This is a listing of the shortcut and command keys available under KSirc.

The KSirc Handbook

Keybinding	Action
Ctrl+N	Open a new Channel/Query Window.
Ctrl+Q	Quit
Ctrl+Enter	Begin a message to the most recent person to send you a message.
Ctrl+Shift+Enter	Begin a message to the last-but-one person to send you a message.
Ctrl+K	Start color code with a number. Text you type after this will be colored.
Ctrl+U	Begin or end underlining text. Text after the first occurrence of this will be underlined until you type the shortcut again.
Ctrl+I	Begin or end italic text. Text after the first occurrence of this will be italic until you type the shortcut again.
Ctrl+B	Begin or end bold text. Text after the first occurrence of this will be bold until you type the shortcut again.
Ctrl+R	Begin or end reverse video text. Text after the first occurrence of this will be reversed until you type the shortcut again.
TAB	Tries to complete the nickname of someone on the channel, based on the text you entered. You are probably familiar with this from your commandline shell, although it does not necessarily work in exactly the same way. See [?].

Table 6.2: Keybindings

Chapter 7

Tips

7.1 Joining KDE channels

KDE IRC channel are hosted on the Freenode network. Use the Server/Channel tab or the Auto Connect tab in Configure KSirc to add these channels. The network should be `irc.freenode.org` on port `6667`. Below are listed the most important KDE channels and their goals. Choose the channels you are interested in.

You can find some tips about asking questions on KDE channels on the [KDE community wiki](#).

Channel	Target
#kde	User help
#kde-devel	KDE development
#kdevelop	KDevelop questions
#kontakt	Kontakt (KMail, KOrganizer, KNotes, ...) related questions
#amarok	Community channel about amaroK: user questions, development
#debian-kde	Debian KDE related questions
#kde-freebsd	FreeBSD KDE related questions

Table 7.2: KDE IRC channels

7.2 Auto identify

If you intend to use IRC on a regular basis you will probably want to register your nickname. This protects your nickname so that only you can use it. On DALnet, after you log on, you register your nickname using the command `/msg nickserv register PASSWORD`, where `PASSWORD` is your password. You'll get a confirmation your nickname has been registered.

Then each time you log on to DALnet you get a message asking you to identify yourself. You will then need to type `/msg nickserv identify PASSWORD`.

KSirc can identify you automatically when you log into any channel. You simply need to add the following Perl script in your home directory:

The following script will work on Freenode. This script should be saved as `.sircrc.pl` in your home directory. When you run KSirc, this script will automatically have you identified.

```
sub hook_nick_notice {
my $n = shift;
my $m = shift;

if($who =~ /NickServ/){
if($m =~ /If this is your nickname/){
$silent = 1;
&msg("nickserv", "identify XXXXXX");
$silent = 0;
}
}
}
&addhook("notice", "nick_notice"); # join on the "end of ←
MOTD" numeric
```

Replace `XXXXXX` with your password.

The above script will work on other networks than Freenode where nickserv sends you the request "identify" but on some networks like Undernet you will need the following script:

```
sub hook_connected {
if($server =~ /undernet.org$/){
&msg("X\@channels.undernet.org", "login YOUR_NAME XXXX");
&docommand("umode +x");
}
}
&addhook("255", "connected"); # join on the "end of MOTD" ←
numeric
```

Replace `YOUR_NAME` with your login name and `XXXXXX` with your password.

You can find more about SIRC programming on this page: <http://www.iagora.com/-espel/sirc/PROGRAMMING>

Chapter 8

Credits and Licenses

KSirc copyright 1997-2002, the KSirc developers.

Portions of documentation copyright 1997, Andrew Stanley-Jones

Documentation updated for KDE 3.0 by Philip Rodrigues phil@kde.org.

chapter 7 was written by Anne-Marie Mahfouf annma@kde.org.

This documentation is licensed under the terms of the [GNU Free Documentation License](#).

This program is licensed under the terms of the [GNU General Public License](#).

Appendix A

Installation

A.1 Where to get KSirc

KSirc is part of the KDE project <http://www.kde.org/>.

KSirc can be found in the kdenetwork package on <ftp://ftp.kde.org/pub/kde/>, the main FTP site of the KDE project.

A.2 Compilation and Installation

In order to compile and install KSirc on your system, type the following in the base directory of the KSirc distribution:

```
% ./configure
% make
% make install
```

Since KSirc uses **autoconf** and **automake** you should have no trouble compiling it. Should you run into problems please report them to the KDE mailing lists.