

Frequently Asked Questions about mIRC.

The mIRC FAQ is maintained by Tjerk Vonck

This is FAQ version 56, revision 0

Latest FAQ Update: March 3rd, 2004

Latest mIRC version: 6.14 March 3rd, 2004

This FAQ attempts to answer the most frequently asked questions about the IRC client mIRC, developed by Khaled Mardam-Bey. This FAQ will only answer simple questions on IRC itself since lots of info is already available on that. (Eventually read the IRC Intro included in mIRC). Not nearly all functions and features of mIRC are explained in the FAQ; it merely serves as a good starting point. With the FAQ I hope to help you to get all possible basic problems solved, and to give you lots of good ideas about mIRC's scripting capabilities. I'm not the author of mIRC but I've been around for years and spent quite some time beta testing it. I love mIRC for its small size, speed and useful tools,... in particular the popup menu's aliases and the scripting section are really great! Thanks Khaled, for an excellent proggy !

This FAQ as well as mIRC is still a work in progress! Most subjects in this file were prompted by questions in the Usenet newsgroups alt.irc, alt.irc.questions and alt.irc.mirc. Recently mIRC dedicated message boards have been added to the mIRC website. These boards offer more support and contribute to the FAQ in a great way! Please continue to post your questions there (and answers too!), as those forums are my learning place as well. This FAQ might not be totally complete yet, but to the best of my abilities I try to be correct. Do not hesitate to provide additional information or corrections for the FAQ.

The first part (Sections 1 - 6) of this file is the actual FAQ. The last part (Section 7) consists of a tutorial or reference manual for mIRC's programming features. If you want to learn the 'what and how' of creating Aliases, Popsups and Remote Commands and Events in mIRC, check out the last part of this file. I can highly recommend these sections to you all !

Thanks to all the people who voluntarily contributed to this FAQ. In particular Mookies, Bryan and Li0nheart for making additional parts and html-ing. Shorty, Keyman and Qasimtoep, thanks for fixing a lot of spelling and grammar bugs! Thanks to Junyor for his contributions from the alt.irc.mirc FAQ

You can e-mail me with FAQ related remarks at: tjerk@mirc.com

Questions about mIRC are best asked in the message boards on mIRC's website

As usual, direct all bug reports to Khaled Mardam-Bey at: khaled@mirc.com but first read his personal FAQ at <http://www.mirc.com/khaled/faq.html>

If you have further questions about mIRC, please visit the IRC channel #mIRC on EFnet, IRCnet, Undernet or Dalnet. BUT, if you ask a question that is really well handled by this or other FAQ's, then please accept that you be pointed back to this or other help files.

This FAQ is also available on the World Wide Web on the mIRC Homepages :

http://www.mirc.com	USA
http://www.mirc.co.uk	United Kingdom
http://mirc.eon.net.au	Australia
http://www.mirc.queen.it	Italy
http://mirc.kems.net	Kuwait
http://www.mirc.co.za	South Africa

Copyrights You are allowed to provide and distribute the mIRC FAQ -as is- by or on any medium as long as you make it available for free. You are not allowed to change anything in the file or charge any amount of money for your services. If you want to copy only certain parts for whatever use, make sure to mention my name and the FAQ as the source of information with every single quote whenever you publish it. Copyright 1995-2004 Tjerk Vonck tjerk@mirc.com

Table of Contents

- 1 [WHAT IS mIRC?](#)
- 2 [LATEST NEWS ON mIRC.](#)
- 3 [WHAT DO YOU NEED TO RUN mIRC? WHERE TO GET IT? HOW TO REGISTER?](#)
- 4 [SHORT INTRODUCTION TO IRC.](#)
- 5 [SOME SHORT NOTES AND TIPS ON mIRC AND IRC.](#)
- 6 [FEATURES, TIPS and ANSWERS to QUESTIONS.](#)

- 6-1 [How do I run the 32-bit mIRC on win 3.1 or WFW311 ?](#)
- 6-2 [How do I run the 32-bit mIRC on Windows95, 98, 2000 or NT ?](#)
- 6-3 [Is there a version for Windows CE, WebTV, Mac or Linux?](#)
- 6-4 [How do I solve the "Can't Resolve Local Host Name" error ?](#)
- 6-5 [How do I solve the "Unable to resolve IRC server name" error ?](#)
- 6-6 [How do I solve the "Not enough user parameters" error?](#)
- 6-7 [How do I solve the "You haven't registered" error ?](#)
- 6-8 [Why am I unidentified and what does it matter ?](#)
- 6-9 [I can't get mIRC to DCC send or initiate a DCC chat !](#)
- 6-10 [I can't get mIRC to DCC get!](#)
- 6-11 [How do I Copy and Paste ?](#)
- 6-12 [How do I customize the colors in mIRC ?](#)
- 6-13 [How can I customize the fonts used in mIRC ?](#)
- 6-14 [How can I select custom backgrounds in mIRC ?](#)
- 6-15 [What is a script?](#)
- 6-16 [How do I make a script?](#)
- 6-17 [How do I specify people correctly in the auto-op, protect and ignore list ?](#)
- 6-18 [What is the best script?](#)
- 6-19 [What are trojans? What are script.ini and dmsetup.exe?](#)
- 6-20 [How can I make mIRC react to Remote Commands ?](#)
- 6-21 [How can I make mIRC react to Events ?](#)
- 6-22 [How do I specify my friend Bill correctly in the User list with an access level of 2?](#)
- 6-23 [I get disconnected after the /LIST command.](#)
- 6-24 [How can I protect myself against flooders?](#)
- 6-25 [Running multiple copies of mIRC.](#)
- 6-26 [How do I make myself invisible to other users ?](#)
- 6-27 [How does the new sound command in mIRC work ?](#)
- 6-28 [How does the WWW support in mIRC work ?](#)
- 6-29 [What is the built-in finger server for ?](#)
- 6-30 [How can I easily serve my collection of shareware to others?](#)
- 6-31 [How do I use mIRC's built in support for Text To Speech by Monologue ?](#)
- 6-32 [What is a Firewall ? How can I use it ?](#)
- 6-33 [How do I use mIRC's built in DDE support ?](#)
- 6-34 [I switched to a new provider but mIRC keeps using the old one !](#)

- 7 [SOME NOTES ON 'PROGRAMMING' in mIRC.](#)

- 7-1 [The Aliases section.](#)

- 7-1-1 [Creating Aliases.](#)
- 7-1-2 [String definitions.](#)
- 7-1-3 [Identifiers.](#)
- 7-2 [The Popups section.](#)
- 7-3 [The Remote section.](#)
- 7-3-1 [Remote Users.](#)
- 7-3-2 [Remote Scripts - CTCP Commands.](#)
- 7-3-3 [Remote Scripts - Events.](#)
- 7-3-4 [Various flags you can use in front of Command or Event Script lines.](#)
- 7-3-5 [Variables.](#)
- 7-3-6 [Remote Scripts - Raw processing.](#)
- 7-4 [Advanced use of commands - Multi-line commands.](#)

1-1 What is mIRC?

mIRC is a shareware IRC Chat client for Windows. It is developed and copyrighted by Khaled Mardam-Bey. For those of you new to the Internet, IRC stands for Internet Relay Chat. The IRC network is a virtual meeting place where people from all over the world can meet and talk (well, type). On IRC you meet others on "channels" (rooms, virtual places, usually with a certain topic of conversation) to talk in groups, or privately. There is no restriction to the number of people that can participate in a given discussion, or the number of channels that can be formed on IRC. As a user you run a "client" program like mIRC which connects to a "server" in an IRC network. All servers are interconnected and pass messages from user to user over the IRC network. One server can be connected to several other servers and up to hundreds of clients. If you need more information on IRC go to mIRC's www Homepages where a lot of additional information is given.

mIRC attempts to provide a user-friendly interface for use with the IRC network. It has the following features:

- A powerful dynamic /help command.
- Full Netscape, Mosaic and MS Internet Explorer support surf the www waves !!
- An 'off you go' toolbar... and even tooltips !
- A very handy drag and drop Switchbar.
- A user programmable menu bar.
- A simple and smart CTCP commands and events handler.
- Support for .wav, .midi and .mp3 sound files.
- Fully configurable colors and backgrounds.
- Simple, fully configurable aliases.
- Configurable popup menus.
- Full DCC Send/Get/Chat support.
- Fully configurable fonts and support of colored, bold, underlined and reverse text.
- Multiserver support.
- Built in Ident and Finger server.
- A simple but powerful and safe built-in File Server.
- Programmable Function keys.

The various parts of the program have been designed with the aim of simplifying and speeding up your IRC sessions. mIRC is made to be very configurable... there are a lot of simple switches you can use to personalize mIRC to your needs. And unlike a lot of other IRC programs mIRC still gets more and more mature every new version. You can define your own commands and implement your personal reactions to commands others give to your client. This all does not make mIRC a bot program, and it doesn't even support IRCii scripting and possibly a large number of other things... But it has much of the same functionality, thanks to the Remote Commands and Events and various other features... and what do you need a

war script for, anyway?

I think mIRC will help you to focus on the main point ; IRC is for Chatting

Distribution mIRC is spread over the world by advanced distribution schemes of primary and mirror FTP sites, which makes it available from hundreds of places all over the world. The most up-to-date list of places where you can get mIRC is the download page on the mIRC Homepages... You could also join the IRC channel #mIRC to get the latest version or to ask all remaining questions... (But: be very sure this FAQ doesn't answer your question)

1-2 mIRC is Shareware.

mIRC is shareware. This system of distributing programs shows the true spirit of the Internet. You can download mIRC for free and give it a try. If during or after the 30 days evaluation period you decide to continue to use mIRC, you're asked to pay a small registration fee of \$20.

So, if you enjoy using mIRC, or find it beneficial, or if you hate it but you like our humble effort, please register your copy. Many years of care and work have gone into making mIRC, and your registration helps to support continued work on it. Don't forget; Khaled is not a million dollar dotcom. He drops dead like you and me when mIRC doesn't pay his food and rent...

mIRC can be registered online. You don't even need a credit card! You can find out how on the mIRC website. The registration process might look a little confusing and offers you a fresh copy of mIRC for download that you might have already. It -will- result in a successful order of your personal registration number for mIRC. Khaled will send it to you personally, by email. He can be reached at khaled@mirc.com with questions or to check up on the status of your registration.

Once you registered one version of mIRC, you're allowed to use all future versions for free!

[Table of contents](#)

2 Latest News on mIRC... (What is new in version 6.14)

This new release addresses the most important comments, requests and remarks received after the release of version 6.12. The new version fixes lots of the small but nevertheless annoying buggies found in the previous version. Among them the much discussed tooltip gpf bug and a daylight savings time offset bug.

While we tried to focus on fixing things, of course some new features and functions have been added. Although some of these new things, like the SSL support, might have to mature a bit over the next versions, we think it is fun to play with them already! A few items I'd like to mention as useful changes:

Lots of work has been done on the support of multi-byte character encodings and the sjis-jis support. Khaled has added a "Multibyte editbox" option to the Messages dialog, which makes mIRC use a richedit editbox in all mIRC Windows for entering text. mIRC was using the standard Windows editbox before. The richedit editbox displays complex scripts properly. The Sjis/Jis and Multibyte options are now automatically enabled if mIRC detects a Multibyte version of Windows the first time mIRC is run. Khaled also fixed various multi-byte display/mark/copy bugs and improved sjis-jis support.

An option to 'reload' logs has been added to the /IRC/Logging/ dialog. It will make mIRC load

the tail of a logfile into your channel and chat windows. Seems to work fine. However, various line attributes, such as color, are not (yet) saved in the log files, so the reloaded log is displayed in the normal text color.

Support for SSL has been added. You'll have to download some DLL's of OpenSSL 0.9.7c separately from mIRC and place them in the mIRC folder for this to work. mIRC will load the dll's automatically if it finds them, after which you can initiate a secure connection to an SSL capable IRC server. SSL settings can be found in the Connect/Options dialog, and are visible only when openssl dlls are loaded. Read more on <http://www.mirc.co.uk/ssl.html>.

A Sort dialog has been added to the Window Menu. It allows you to specify the sort order of Switchbar buttons and Window types. Buttons are always sorted into groups. With the /View/Options/Display/ sort option turned off, Switchbar buttons are sorted according to creation time within groups as opposed to alphabetical sorting.

By popular request Query/Chat windows now remember their individual font settings. We also

- moved the port range options, which now applies to all connections, to a /Connect/Advanced/ dialog,
- added a /cnick -sN switch, sorts item into Nth position,
- made the switchbar to allow up to eight lines in height and displays scroll buttons when it is vertical,
- fixed \$com() dispatch pointers bug,
- added an :error goto point, by which script errors can be caught, and
- improved the find dialog in the script editor a lot.

As you see there are heaps of changes and improvements. Lots of little and larger changes to smoothen your IRC experience. You'll have to dig through the help file in mIRC and the whatsnew.txt available on the mIRC website to learn about the complete list of changes and their impact. Some are obvious, some need getting used to - please take your time to play with them and see how it works. If you have questions, may we invite you to the Message Board on the website? This board offers great help with everything related to mIRC!

What was new in version 6.12? (October 13th, 2003).

mIRC v6.12 has been released to address a remote vulnerability found October 12th, capable of crashing your mIRC. The vulnerability affects versions of mIRC from v6.0 onwards, so it is highly recommended that you upgrade to mIRC v6.12. You can do so by downloading the new mIRC from the download page on mIRC's website!

This version is merely a bugfix version. Although no new features have been added it is very important that you upgrade as soon as possible. If you have questions, may we invite you to the message board on mIRC's website? These discussion forums offer great help with everything related to mIRC and its features.

What was new in version 6.11? (October 10th, 2003).

With this new release we hope to address the comments, requests and remarks we received after the release of version 6.1. Of course this new version fixes lots of the small but nevertheless annoying buggies found in the previous version.

Several commands have been fixed or improved, like the /fwrite, /hdec, /mkdir, /qmsg, /qme, /color, /hload, /hsave and /server not using the specified port. Lots of identifiers have been fixed, among them \$duration(N,3), \$base(), \$os, \$comcall(), \$dllcall(), \$md5() and \$mask(). This also fixes a possible /userhost exploit problem. A channel folders join bug where mIRC

wasn't setting focus on a channel window properly was fixed, as well as a small problem with the red dot in remote toolbar button, bugs with binary variables, duplicate separators in popup menus and a menu handling bug that affected the favorites popup menu as well as various other menus under Windows 95 and NT. Over 40 thingies, all fixed :)

While we tried to focus on fixing things, of course some new features and functions have been added. Among them

- the language selection option that has been added to the MS Agent speech dialog. You must of course have the required language component installed. Read more on <http://www.mirc.co.uk/agents.html>,
- the Favorites folder that now allows multiple selection. You can now join multiple channels, or folders with channels!, at a time,
- a nifty new feature that has been added to allow you to auto-hide the list with nicknames in a channel. See the nick list options in the System Menu of each channel...

What was new in version 6.1? (August 29th, 2003).

Lots of things have been fixed. Like the default browser issue, a /background bug, an editor dialog bug that most likely was the cause of odd script problems, a memory bug relating to registry handling, a /whois related memory bug, a problem with background pictures, and a popup menu resources bug.

Lots of other things have improved; All Socks5 firewall error messages are now displayed correctly, double-clicking on web/ftp hotlinks now displays a warning dialog before the site is opened, and an important change was made to the way files are created, opened, read, etc. for window logging, dcc sends/gets, and the list channels dialog. The amount of files opened for logging, sends/gets, etc. is no longer limited to a maximum total of 50 or so open files. DCC commands, such as /dcc send, ignore, etc. are now also handled in the form /dccsend, /dccignore, etc. and should also allow /dccallow etc.

Khaled is now using Visual C++ .NET 7.0 to compile mIRC. This also means nicknames are sorted a bit different, nicks beginning with certain non-alphabetic characters eg. _ ' etc. are being placed above normal nicks. This appears to be the standard sorting method in Windows, which Visual C++ abides by consistently. It now applies to the whole of mIRC. Support for the .jpeg extension has been added, custom file-handling commands and identifiers which allow you to open multiple files for reading and writing. A file type ignore warning explains why a file transfer is ignored, with a direct link to the options dialog, dcc ignore section. Also a new section was added to the servers.ini file from which mIRC will select an initial network to connect to. mIRC no longer defaults to DALnet.

Lots of interface changes this time; Check out the visual styles and new buttons, dialogs, menus etc. A single toolbar button replaces the previously separated aliases, remotes and popups button. The red light is now on when any of ctcp/event/raw are enabled, and off when they are all disabled. A right-click on the toolbar button displays a most recently used files list. At first this looks a bit odd, but it will just take some getting used to. With the color scheme option you can switch between different color schemes easily.

And scripters, there is lots of fun for you; with the \$ event prefix the matchtext section of a remote event definition can now contain a regular expression! Binary variables no longer have an upper limit on the number of bytes that can be stored. And \$compress() and \$decompress() functions have been added that compress/decompress files and binary variables. The result is raw compressed data. Khaled managed to load a multi-megabyte file into a binary variable and compress, encode and write it to a new file, and then reverse the process. Fun stuff eh!

- The "enable" sounds option in the sounds dialog now affects most sounds, including highlight, notify, window/icon flashing, etc. Sound requests and Agents still need to be enabled/disabled individually. This does not affect the use of sound-related commands in scripts.
- Pressing Control-C while marking text in a window now copies the text and cancels the mark process. It took eight years to think of that. It's all about quality you know ;)
- Remember how you can 'collapse' all windows you have opened on a network into their status window with Alt+F1? The status window switchbar button now displays the highlight state of all associated windows hidden with Alt+F1.
- Disconnecting via the file menu/toolbar/disconnect command now results in a soft disconnect, allowing any pending server info to be displayed. Incoming data will not trigger scripts, etc. during this time. Eg. if you connect to EFnet, and then disconnect, EFnet sends a "Closing Link" message to the client. Past versions of mIRC wouldn't display that since they were performing a hard disconnect.

What was new in version 6.03? (August 16th, 2002).

The 'perform on connect' sorting mistake has been fixed, also the numeric 330 bug, the flickering menubar and toolbar, the ! on INPUT processing bug, the \$read and a similar freeze bug with invalid filenames, a switchbar multi-line display bug, dcc ignore bug, a /timer bug, and a bug in the highlight feature are fixed. The \$ticks has changed back to the old method because of a bug in Windows' high performance timer... *mumble*

Other fixes are the /load command that could trigger multiple on load events, a proxy authentication bug, the \$asctime() bug, a small dcc server ignore bug, and an important memory bug in script parser, where mIRC checks if a script has been loaded/unloaded while a script is running.

mIRC now supports several IRCX events on any server that sends them, not just for IRCX servers, it will (again) try to rejoin a channel even if it was +i or +k, we improved the way mIRC determines what default web browser you use, rewrote the mark/copy routine, and added a \$nick().idle property that returns the idle time for a person on a channel. You can also use this idle time feature in the nick color list dialog in the addressbook!

What was new in version 6.02? (June 6th, 2002).

With this new release we hope to address most, if not all, of the comments, requests and remarks we recieved after the release of version 6.01. Several changes are initiated by the multi-server features introduced in mIRC 6.0. You might want to check out how the 'perform on connect section' has been moved to the connect section, near the server settings, and allows you to specify a different set of commands for different networks, and how basic features as the /aop /avoice /protect and /ignore commands now allow you to specify a network name.

The multi-line switchbar options in the display dialog have been extended, mIRC no longer closes a channel window on reconnect to a server if you have keep channels open enabled, no longer attempts to rejoin a channel on reconnect if the channel was invite only or if it was locked and mIRC didn't have the key.

Changes scripters will like include the \$ticks which does no longer wrap around every 49.7 days, custom dialogs that no longer require an ok or cancel button, and new events like on CONNECTFAIL that triggers when a connection attempt including its retries has failed.

Fancy new things have been added like the Alt+F1 key combination that hides all windows associated to a status window switchbar button 'in' that button. Try it and you'll see what I

mean!

Several more or less annoying bugs have been fixed

- A memory handling bug which may have caused mIRC to crash in situations where it wasn't able to allocate memory,
- a bug with mIRC not saving certain settings on exit, including /notify command line changes, as well as server settings, ignore thingies, etc,
- a tab key nick completion bug,
- a window tiling bug when a maximized window was closed while auto-tile/cascade was enabled,
- the on keydown/keyup events in custom @windows not handling some keys,
- editboxes that limited the amount of text that could be entered,
- the "* Disconnected" message not being displayed in channel window in some situations,
- a problem where mIRC was not using the system default for browser/email programs,
- a small problem where the nick in a channel/query window titlebar not being updated on a nick change if the "Show nick in channel/query titlebars" option was enabled,
- ...and lots of other things are improved or fine-tuned after user feedback, your feedback!

What was new in version 6.01? (February 10th, 2002).

Version 6.01 addresses most, if not all, of the comments, requests and remarks we received after the release of version 6.0. We'd like to emphasize the new visual style dialog that allows you to set and change frame styles (borders, lines and bevels), switchbar buttons, and a new option to disable support for irc:// chat links. By default you also have to confirm chat link requests btw.

Several more or less annoying bugs have been fixed; a socket bug which that very likely caused the crashes some of you have been reporting, a connection retry bug which may have resulted in wrong nicknames, channel prefixes and nick mode prefixes, etc. Other fixes are; an /ignore -u bug with multi-server support, the password editbox, Agent support, IRCX support for MS servers, the on NOSOUND event not triggering, the notify list that got lost if you used command line parameters and dcc send/get windows not showing "transfer incomplete".

What was new in version 6.0? (February 3rd, 2002).

As usual the new MIRC offers heaps of changes and improvements. Lots of little and larger changes to smoothen your IRC experience. Since mIRC 6.0 is no longer available in a 16 bit issue all references to the old 32 and 16 bit distributions have been removed. Lots of 16bit-specific code has been removed from mIRC's routines. Lots of 16 bit related memory issues, problems, kludges and bugs are solved. All dialogs in mIRC should now use standard dialog fonts for the version of Windows you are using. When running XP, mIRC now uses your XP theme for dialogs, buttons, etc....

Some of the many many improvements, changes and fixes in this new mIRC are

- The max cps fserve option now allows a dcc send to use up the full max cps if no other dcs are using the bandwidth.
- You can now create a second editbox on channel windows, via the channel window system menu, the right-click menu in the editbox, or the Alt+Q shortcut.
- mIRC will no longer rejoin an open channel window if you disconnect and then connect to a different network.
- Added WMA/OGG support to sounds dialog, /splay and /sound commands, and added several new new identifiers to support it. (Note: these new identifiers replace several old ones that are still supported -for now- but no longer documented.)
- Added 'Trusted Users' list to DCC Auto-get options dialog.

- Firewall support can now be enabled separately for server and dcc connections.
- When running under XP, mIRC now uses your XP theme for dialogs, buttons, etc.
- mIRC now processes incoming server messages rather strict, ignoring corrupted or malicious lines, and treats ctcp messages that use an invalid format as plain text.
- Added "Queue own messages" option to flood dialog, applying to your PRIVMSG and NOTICE messages.
- The connect retry options (including a new retry delay option) in the File/Options/Connect/Options dialog now have their own button.

Multi-Server support has been added. With this feature mIRC allows you to connect to multiple servers at the same time. You'll notice a new File/Options/Connect/Options/ 'Multi-Server' button that provides a couple of new options supporting this new feature. The new multi-server feature required major internal changes to mIRC's code and forced improvement of code in many areas which has undoubtedly resulted in an even more stable and robust mIRC! Working on this important new feature we focussed on keeping mIRC as it always has been; no-nonsense, robust, -working- :) We tried to keep the changes to the scripting language etc, as clean as possible. Upgrading should be fairly painless. Still a multitude of ID's, identifiers and commands has been changed and added to support the new parallel connections. Everything in mIRC now has a unique id number; all windows, server connections, dcc sends/get/chats, etc....

What was new in version 5.91? (June 15th 2001).

Among the buggies you might have noticed is one that messed up the merging of popup menu's. In this new mIRC complete menu's from different scripts will merge properly. Also fixed are bugs in \$eval(), \$var() and small thingies in the 16 bit mIRC. A problem with the /whois on query option not working in single message window is solved. Overall there aren't many things you'll notice but all together enough to make your upgrade worth the trouble.

Scripters should note that \$null has been completely removed as a value. This has some potential to break your scripts. When you use \$null in a script, mIRC now converts it to an empty value, instead of the value "\$null"!

- The flood protection has been improved with options to take your own ctcp's and 'whois on query' settings into account. You can even make Op status related commands (like /kick) to bypass the anti-flood queue.
- Since IRC networks don't provide a consistent method of telling an IRC client the name of the network it's connected to, mIRC now supports the numeric 005 NETWORK=name token. Hope that helps?
- When you have the "keep channel open" option enabled, mIRC now clears the nicklist if a channel window is open but not joined.
- A "Blink icons" option is added to the Display dialog. With this option the switchbar icons of channel/query windows will blink if there's a message/highlight or flash event.
- Notify no longer triggers on a nick change if nick has only changed case.
- A \$(n1,...,nN) identifier that combines parameters has been added.
- The notify options to show address and time are re-added.
- The Hash commands are extended to allow storage of binary variables.
- The variable types dispatch and unknown are added to \$com(). These allow you to pass dispatch/unknown pointers as parameters in a \$com() call, or to retrieve dispatch/unknown pointers from a \$com() call, by reference.
- You can now pause and resume a timer with the /timer -pr switches.
- mIRC now waits 30 seconds before a re-connect attempt if the server says that you were throttled for connecting too fast.

You know by now that you have to dig through the help file and versions.txt to get hold of

the complete list of changes and their impact? Don't forget that ;-) If you have questions, may we invite you to the Message Boards on mIRC's website? This discussion platform offers great help with everything related to mIRC!

What was new in version 5.9? (April 26th 2001).

As usual the new MIRC offers heaps of changes and improvements. Lots of little and larger changes to smoothen your IRC experience. In fact, this version has far too many improvements, new features and functions to mention here in detail. The complete list is included with mIRC and available on the web.

- Something you should really take a look at is the new and improved Address Book. It still offers an easy interface to addresses and related info of other IRC users but now also includes your notify list, Op, Voice, Ignore and Protect lists and a new feature to colour all nicknames around that match a certain format. You wanted all ops blue? your friends green? target and enemy red? You got it!
- A smart "Keep channels open" option has been added to the IRC/Options dialog. It keeps channel windows open after you're kicked so you can read the last part of a conversation.
- Built-in ctcp flood protection for ctcp version, ping, finger, and time has been added. It will ignore (groups) of users for up to 10 seconds.
- In the display section you can now enable transparency support for desktop windows under Win 2000.
- The dcc send/get windows interface has been redesigned. You can now easier open folders and resend files straight from the dcc send dialog. Also a small error in the DCC Send/Get cps calculation was fixed and the maximum bandwidth use by outgoing DCC's is better limited now.
- Agent support no longer strips out high-ascii characters used in non-english languages.

All you mIRC scripters will love this version for its load of new features. You know by now that you have to dig through the help file and versions.txt to get hold of the best among them? Don't forget that ;-)

You should note that several Strings and Identifiers have been changed! Most of them have more options or parameters or are changed to fit better to the standard.. No worries; most old formats are still supported but you -should- update your scripts to prevent problems in the future!

- We think most of you will be happy with the simple things like \$input(), \$read() instead of \$read, and \$style() for popups. Also \$eval() is nice, and please note the parser change for [brackets]'s!
- The script editor editbox can now handle 64k of text in the 32bit mIRC.

Things might get a little tricky when you try the new support for regular expressions (regex), COM objects and SendMessage() to communicate with mIRC. This really is something for the hefty scripter...

What was new in version 5.82? (December 14th 2000).

- The support of the irc://address.or.server and .chat file formats has improved a lot. It will use DDE again so if a mIRC is already running on your PC, it is used. If no mIRC is found via DDE, a new mIRC is opened. It pops up a confirmation dialog if mIRC is already running and connected to a server. Read more on <http://www.mirc.com/mirclink.html>
- By popular request the Names button is added back to the Channels Folder.
- Try the new Control+L key combination. It will enable a line that marks the last line in the scrollbar buffer of a window that you most probable have read. The line is updated if mIRC isn't the active window. Control+L turns the line on/off if it's currently visible.

- Mousewheel support for microsoft mice is fixed. Some small thingy blocked its use in previous mIRC versions. A small bug in the DCC ignore settings dialog is fixed. Also the channel central, that was trying to set topic even if you didn't have ops, is fixed. You can open the channel central with the /channel command btw.
- The 32bit mIRC will no longer work under Windows 3.1x with the Win32s 32-bit extentions. Since the 16 bit mIRC offers exactly the same as the 32 bit mIRC this makes no real difference to users of old Windows versions. Simply use the 16 bit mIRC. This change shaved 100k off the exe size and makes mIRC faster since no internal fixes are being used any more.

What was new in version 5.81? (November 9th 2000).

- Remember how mIRC now uses it's own internal beep sound for event beeps? By popular request the internal beep is now only played for event beeps. mIRC uses the default windows sound for errors etc. and a "pc speaker" option has been added overriding both of them. Happy?
- Another loong awaited feature is a built-in uninstaller. I wouldn't know why that is needed but next time you clean up your PC, mIRC is easily removed by the control panel add/remove dialog! It also removes all registry changes btw.
- An important change has been made to the socket connection routines. mIRC was binding the connect socket, even if it wasn't necessary. This may have prevented mIRC from connecting through a DSL or network environment.
- The Flood protection has been improved by sending a NOTICE instead of a PRIVMSG to the server every now and then. As you might know mIRC will wait for this notice to get back from the server before it resumes sending any queued lines.
- A small problem with the .chat files that unloaded all of your remotes has been fixed. More important is the complete integration of .chat file support into the mIRC executable. The mlink helper applications are no longer needed! mIRC will also set up the .cha and .chat associations in the registry.
- With mIRC 5.81 we're experimenting with support for URL's of the irc://irc.undernet.org:6667/mIRC format. This makes it very easy to add a link to your website leading to your favorite IRC channel! Give it a try? Please note however that this only seems to be working for MS Internet Exploder -not Netscape- we're still working on this :) See <http://www.mirc.com/mirclink.html> for up-to-date info.
- The new /anick command no longer changes your current nickname, unless the alternate nick is the active one. /nick will try to prevent nick and altnick from becoming identical nicks.
- By popular request you can now either disable, or ignore all except certain known file types, or ignore only certain known (and often malicious) file types in the DCC Ignore feature.

Buggies fixed: mIRC's new version fixes most, if not all, of the small but nevertheless anoying buggies found in the previous version 5.8. Some prominent ones are

- A small change in the Channel Central (use the /channel command) blocked the use of control characters in the topic editbox. This blocked the use of colours etc. Sorry.
- The /splay command will now accept long file names even if they're not enclosed in quotes and error messages are now also wrapped.
- A little bug that emerged if you double-clicked URL's under Windows ME is fixed.
- Another buggy in the Agent wasn't allowing you to turn agent on/off on a per channel/query window basis.

Scripters: All you mIRC scripters will love this version for its load of new features. You know by now that you have to dig through the help file and versions.txt to get hold of the best among them. Some things you might miss there are the old /closemsg command since the /close handles closing query windows a-ok. Also gone from the documents are the \$sdir, \$file, \$file, and \$dir identifiers that are still supported but are no longer documented in the

help file. Please remove their use from your scripts and use their new alternatives! In the future mIRC will no longer support them.

I'm sure you will have a lot of fun! Apart from removing thingies, about hundred other features, variables and identifiers are improved or added. Have fun exploring them! Please read the details of all changes in the versions.txt appended to this file, distributed with mIRC and available on the Web. Good luck and have fun with this new mIRC!

What was new in version 5.8? (September 5th 2000).

Buggies fixed: mIRC's new version fixes most, if not all, of the small but nevertheless annoying buggies found in the previous version 5.71. Some prominent ones are a bug in the Agent support that made Agents speak channel actions even if only private actions were enabled, a bug in the highlight feature which was matching against nicknames incorrectly in some cases, and a small bugger in the flood protection.

Improvements and new features: This version has far too many improvements, new features and functions to mention here in detail. I'll point to some of the simple ones here - the ones you might encounter right away;

- mIRC now uses it's own internal beep sound for event beeps. To several users it was very annoying that some setting in windows messed up the most simple beeps in mIRC. This should be solved now.
- The URL's you can click on in mIRC now have their own right-click popup menu.
- The "No such nick/channel" message is now immediately shown in query windows. You won't be chatting into empty space anymore if somebody decided to leave without telling you.
- A new channel central dialog displays the ban, exopt and invite lists a lot clearer. Also an Edit button has been added for on the fly editing of an address.
- mIRC now supports mp3 files through the /splay command. Several related events and identifiers are available. With the \$mp3(filename) identifier you can look up all kinds of properties like title, artist, length, bitrate, sample mode, etc..

Scripters will love this version for its load of new features. You might like to know mIRC can now be run as a service under 95/98, you can change the rgb value of a colour, use identifiers to mime or uu-encode/-decode text, use the improved support for DLL's, use hash tables.... I'm sure you will have a lot of fun!

Apart from these thingies over hundred other features, variables and identifiers have been added. Have fun exploring them! Please read the details of all these changes in the versions.txt distributed with mIRC and available on the Web on <http://www.mirc.co.uk/versions.txt>. Good luck and have fun with this new mIRC!

What was new in version 5.71? (May 7th 2000).

- Support for dual monitor displays (in the 32bit mIRC) has been improved by a new option in the Display/Options/ dialog.
- The 32 bit mIRC now supports .JPG and .PNG image files next to the older bitmap support for backgrounds etc.
- To prevent you from getting flooded the CTCP Version requests are now queued by mIRC, and the replies are sent once every few seconds.
- When clicking the "Connect" button to connect to a server, you can now hold down the Control key to force mIRC to use the next server in the list. mIRC has lots of handy functions like this. They are explained in the help file. Some need a little practise, others will come to you by nature.
- Several things in the Channel window interface changed. The number of users in channel is

now shown in the channel titlebar, you can now resize the nicknames listbox in channel windows, you can add colour to the nicknames that are talking (the "Highlight nicknames" option) and to complete things you can now prefix nicknames with their mode on the channel (.@%+) by the "Show mode prefix" option in the IRC dialog. This allows quick insight in the users status.

- After manual support for the Microsoft Agent was added in version 5.7 (see <http://www.mirc.co.uk/agents.html>) built in support has now been added to mIRC in the options/Sounds/ dialog. You can easily enable agent events for channels, messages etc. without any scripting knowledge. It has never been easier to listen to IRC!
- An important change was made to the DCC "ignore file types" feature in DCC/Options/. It now works as an "ignore all except.." filter. This will prevent you from downloading unknown but malicious files from strangers. A simple ignore timer, allows you to turn off the ignore for a small period after which mIRC auto-activates it again.
- When using wildcards the highlight method now matches the wildcarded text against individual words separated by spaces, instead of against the whole line.

Scripters can now have unlimited controls per tab section in a custom dialog and the custom dialog tab control now automatically adds a scrollbar if the tabs don't fit the width of the tab control. mIRC now maintains an internal banlist for each channel and several related strings have been added. Check out the \$banlist for instance! Apart from these lots of other variables and identifiers have been added. Have fun exploring them!

What was new in version 5.7? (February 2nd 2000).

- A new, fresher interface, cute buttons - brighter colors!
- Support for the Microsoft Agent has been added to the 32 bit version of mIRC under Windows95 or greater. An agent is an animated character that can speak text and perform actions. With a little fiddling and tweaking you will be able to have this agent speak to you, warn you about things happening on IRC, or even read-up entire discussions to you! See <http://www.mirc.co.uk/agents.html> for more info. Also, if you have Speech Recognition software installed, mIRC can be made to listen to voice commands!
- A network listbox has been added to the connect dialog. You can now easily select a network and then quickly select the desired server within that group of servers. By default a selection of all random servers is shown. It is now also possible to specify a range in the Ports settings of an IRC server, eg. 6660-6669. This will spread the load on the IRC servers ports a lot more, thus giving faster access!
- The mIRC Installer now sets up support for .chat files in your registry file. Too many new users had difficulties in setting this up, we do it for you now :)
- You can now right-click on the alias, popup, remote and DCC toolbar buttons to pop up the new Quick Access menus. This will improve access to frequently used settings a lot. The channel central dialog (/channel) now uses a tabbed dialog to accomodate support for the new +e (ban exeption) and +I (invites) channel modes as found on IRCnet. With the new 'Display' tab in this dialog you can redirect or disable all sorts of channel event messages. This allows you to see the join, part, quit, mode and other messages exactly where you want them, or not at all!
- Yiihoooo - you can now completely disable the "ping? pong!" messages in the File/Options/IRC dialog.
- mIRC now allows you to select a font script for a font in the font dialog. This is needed to support arabic, cyrillic and chinese characters. People who speak these languages will understand how this works :)
- When you're DCC Sending a lot you might like the new option that sets the "max cps per user" in the File/Options/DCC/Fserve dialog. This allows you to limit the send speed used by a DCC Send to a user in a Fileserver, but is also applied to /dcc sends initiated in a remote script.
- We changed the Identd server behaviour back to how it worked before version 5.61. Not all

IRC servers liked the (correctly implemented) new communication and considered you unidentified.

- Various other small changes have been made; you can automatically sort incoming files by nickname into folders, restart logfiles per day, week or month, and set a completely custom timestamp format for all messages, logs, etc.

- Really lots and lots of scripting things have been added. A bulkload of Identifiers, Dialog improvements and a lot more. Dear Scripters; READ the versions.txt included with the new mIRC for all info and crawl through the help file for the details. Watch out for the new %helper mode, it is easily confused with variables since % is also the %variable prefix! Enjoy the support for while loops that has been added. (This repeats a loop while some expression is true) Multiple while loops can be embedded. You can use /break to break out of the current loop. You can also use the /continue command to jump to the beginning of the loop. Have fun! Oh, and beware of old scripts; We finally removed support for the very old \$parm and *N identifier format!

Older mIRC versions.

The full list of changes has become too large to include in the FAQ, so only the version information for recent versions is included above. You can however still download the full versions.txt file from the mIRC website at <http://www.mirc.co.uk/versions.txt>

[Table of contents](#)

3 What do I need to run mIRC ? Where do I get it ?

Besides running MS-Windows you need to have an Internet account and a properly installed Winsock. If you can use FTP, E-mail, News or other Internet programs from within Windows already, you can safely assume you have both. If you do not have your Internet access properly configured on your PC you should deal with that first.

mIRC is spread over the world by mirror FTP sites, which makes it available from hundreds of places all over the world. Some of the major download sites with mIRC are :

Papa	ftp://papa.indstate.edu/winsock-l/Windows95/IRC/
	ftp://papa.indstate.edu/winsock-l/winirc/
Tucows	http://www.tucows.com/
CWSApps	http://www.stroud.com/
CNet	http://www.download.com/
ZDnet	http://www.zdnet.com/swlib/
MyDesktop	http://www.mydesktop.com/

The most up-to-date list of places where you can get mIRC is the Download page on the mIRC Homepages... You could also join the IRC channel #mIRC to get the latest version or to ask all remaining questions... (But: be very sure this FAQ doesn't answer your question)

mIRC is shareware.

As said, mIRC is a shareware program. This system of distributing programs, as shareware, shows the true spirit of the Internet. You can download mIRC for free and give it a try. If during or after the 30 days evaluation period you decide to continue to use mIRC, you're asked to pay a small registration fee. This will allow Khaled, mIRC's author, to go on developing and supporting mIRC with the same spirit and enthusiasm as he did over the past. Your registration of mIRC will allow the existence of mIRC's www pages for help, hints

and support and further development of mIRC, its help files and the FAQ! Once you registered one version of mIRC, you're allowed to use all future shareware versions for free. The mIRC help file and the Web site give you all information needed to register mIRC.

You can register On-line with a credit card.

You can register mIRC online. Take a look at <http://www.mirc.co.uk/register.html> and see it is all pretty simple. Just have your credit card ready and follow the instructions. You'll receive a registration password by Email within some days from the moment your payment has been processed, from Khaled himself!

[Table of contents](#)

4 Short Introduction to IRC.

This is just a short introduction to IRC. Read more in the IRC Intro file included in the mIRC package! In mIRC just type the /ircintro command to read it.

What is IRC - IRC stands for "Internet Relay Chat". It was originally written by Jarkko Oikarinen in 1988. Since starting in Finland, it has been used in over 60 countries around the world. IRC is a multi-user chat system, where people meet on "channels" to talk in groups, or privately. There is no restriction to the number of people that can participate in a given discussion, or the number of channels that can be formed on IRC. All servers are interconnected and pass messages from user to user over the IRC network. One server can be connected to several other servers and up to hundreds of clients. Several larger and smaller IRC networks exist.

Language - The most widely understood and spoken language on IRC is English. However, as IRC is used in many different countries, English is by no means the only language. If you want to speak some language other than English, (for example with your friends), go to a separate channel and set the topic to indicate that. Similarly, you should check the topic when you join a channel to see if there are any restrictions about language. On a non-restricted channel, please speak a language everybody can understand. If you want to do otherwise, change channels and set the topic accordingly.

Greeting - It is not necessary to greet everybody on a channel personally. Usually one "Hello!" or equivalent is enough. Also, don't expect everybody to greet you back. On a channel with 20 people that would mean one screenful of hellos. It makes sense not to greet everyone, in order not to be rude to the rest of the channel. If you must say hello to somebody you know, do it with a private message. The same applies to good-byes. Also note that using your client's facilities to automatically say hello or good-bye to people is extremely poor etiquette. Nobody wants to receive autogreets. They are not only obviously automatic, but while you may think you are being polite, you are actually conveying yourself as insincere. If some body wants to be autogreeted when they join a channel, they will autogreet themselves.

Behaviour - Remember, people on IRC form their opinions about you only by your actions, writings and comments, so think before you type. If you use offensive words, you'll be frowned upon. Do not "dump" (send large amounts of unwanted information) to a channel or user. This is likely to get you kicked off the channel or killed from IRC. Dumping causes network "burps", causing connections to go down because servers cannot handle the large amount of traffic. Other prohibited actions include:

* Harassing another user. Harassment is defined as behavior towards another user with the purpose of annoying them.

- * Annoying a channel with constant beeping. (Therefore most clients cannot beep at all)
- * Any behavior reducing the functionality of IRC as a CHAT medium.

How to join IRC - The first time you run mIRC you have to fill in some information about yourself (your real name, email address, nickname, IP address and Local Host name) under File/Setup/IRC_Servers and Local_Info, as well as the IRC server with which you want to connect. It's usually best to connect to a geographically close server. When you're new to IRC just pick a server from the prefab list. On IRC you are known to others by a nickname. You are free to choose any nickname you like, up to 9 characters long. Do not use spaces and avoid unusual ASCII characters in your nickname. It is possible you find people that use the same nickname and you may be asked to switch nicknames to avoid confusion.

Getting started - To join conversations, send private messages, and to handle and control mIRC you need to learn some simple commands. All commands start with a forward slash, the "/". Anything that does not begin with "/" is assumed to be a message to someone and will be sent to your current channel, or to the person you are chatting with in a private chat (see below). This list with the most used commands on IRC is carefully explained in the mIRC help file.

/HELP	shows general help or help on the given command
/LIST	lists all current channels
/JOIN	to join a channel
/PART	to leave a channel (same as LEAVE)
/QUIT	exits your IRC session, (same as BYE and EXIT)
/NICK	changes your nickname
/AWAY	leaves a message saying you're away or not paying attention
/WHOIS	displays information about someone
/INVITE	sends an invitation to another user
/KICK	gets rid of someone on a channel
/TOPIC	changes the topic of the channel
/ME	sends anything about you to a channel or QUERY
/MSG	sends a private message
/QUERY	starts a private conversation

Finding your way on IRC - To join a channel, type /join #channelname. Try "/join #irchelp" or "/join #mirc" to give it a try... That's it! Once you get to the channel, you will see people talking. It will probably look like this:

```
<John> Hello Pat, are you new to IRC too ?
<Jake> I dont agree there :-(
<East-r> Can sb give me that too ?? I couldnt find it before
* East-r smiles
<Pat> Nope, I just have a simple question... I think...
```

Note that you will often come in during the *middle* of a conversation. Unless you're familiar with the channel you may want to sit and watch it for a minute or two to see what the conversation is about. Often the channel name (for instance, #Twilight_Zone) has nothing to do with what conversation goes on on the channel (#Twilight_Zone does *not* have discussion about the TV show "Twilight Zone"). So if you join #baseball, don't be surprised if you hear about the SuperBowl picks or even the Rock-n-Roll Hall of Fame Museum! To start talking, just type! And when you're done saying what you have to say, just hit the [return] key. You can start with something simple like "hello!". You don't have to type <nickname> hello! because IRC will insert <nickname> before all of your channel messages. In the channel's title bar you will see the channel's name and perhaps its topic. If you choose to leave a channel, just type /part #channelname

In the channel window that opens once you join a channel you'll see an alphabetical list of people that are on the channel on the right side of the window. Some of them have a @ in front of their name to point out they are the channel operators. A Channel Operator is someone who has control over a specific channel. A Channel Operator can also decide if control is shared or not. The first person to join the channel automatically receives Channel Operator status. Channel operators are the 'rulers' of a particular channel. This means they can kick you out of their channel for any reason. If you don't like this, you complain to them or start your own channel and become a channel operator there yourself.

Read the help - As soon as you joined your first channels and spent some time on IRC you will see there are a lot more commands and possibilities for you to discover on IRC. You might want to read the full version of this IRC Intro on the mIRC www pages. Almost all specific capabilities of mIRC are explained in the help file that came with the package. You just have to do the reading. :-)

Books about IRC and mIRC.

(More information on mIRC's web pages at <http://www.mirc.co.uk/books.html>)

O mIRC sem segredos - Ivan Borba
© 1997 Brasport - Livros e Multimidia - Rio de Janeiro - Brazil
ISBN: 85-85840-62-5

Learn Internet Relay Chat - Kathryn Toyer
© 1997 Wordware Publishing, Inc.
ISBN 1-55622-519-9

Using Internet Relay Chat - Marianne Pyra
© 1995 Que Corporation
ISBN 0-7897-0020-4

The Irc Survival Guide - Stuart Harris
© 1995 Peachpit Pr
ISBN: 0201410001

[Table of contents](#)

5 Some short Notes and Tips on mIRC and IRC in general.

In a humble effort to provide you with all information on IRC you could ever need (?) we made the section below with links and references to additional information available on the Web. The information below (and more!) is already available on the mIRC website (in the 'More Info' section) but we felt the information there was easily overseen. We encourage you to use the links below to investigate for the answers of any additional questions you might have after reading the mIRC FAQ.

All of the pages mentioned below are maintained by IRC enthusiasts. You should be aware that parts of this section will be outdated the moment you read it. All information is as up-to-date as possible and most of it even suits IRC as it currently is rather well ;)

Translations of the mIRC Homepage in other languages :
Español <http://web.jet.es/mirc/>

Additional mIRC info provided by mIRC addicts and lovers :
paiRC mIRC source <http://www.pairc.com/>

The mIRC Resource Center <http://www.mIRC.org/>
#mIRC Help <http://www.mirchelp.org/>
Nazaret's mIRC Place
<http://leonardo.spidernet.net/Copernicus/831/mirc/home.html>
mIRC at DumGuy's place <http://www.nuthin.nu/>
Karen's pages <http://www.netreach.net/~hawk/>
mIRC Etc. <http://www.jps.net/remarshall/>
The world of mIRC <http://www.geocities.com/TimesSquare/Arcade/4869/>
IRC Works <http://www.ircworks.com/>
#New2mIRC <http://w3.one.net/~kidz/new2mirc/>

More mIRC info in other languages :

Deutsch <http://people.wiesbaden.netsurf.de/~sven/irc/mirc.html>
Francais <http://www.creaweb.net/irc/>
Italiano <http://www.mondito.it/bowldc/mirc1.htm>
<http://www.mircscripts.com/mirc-ital.html>
Nederlands <http://www.mircscripts.com/mirc-dutc.html>
Portuguese <http://www.elogica.com.br/users/maligno/mirc/>
<http://www.elogica.com.br/vircio/mirc.html>
<http://www.mircscripts.com/mirc-port.html>
Svensk <http://www.mircscripts.com/mirc-swed.html>

mIRC FAQ in other languages :

mIRC FAQ in French <http://www.multimania.com/arn/irc/faq.shtml>
mIRC FAQ in German http://www.cenobyte.de/irc/m_faq_de.zip
mIRC FAQ in Italian <http://www.panservice.it/simplicity/mircfaq.html>

mIRC's IRC Intro in other languages :

Nederlandse IRC Intro <http://irchelp.huizen.dds.nl/nl-intro.htm>
IRC Intro pa Norsk <http://members.xoom.com/siljel/irc.html>

Other IRC Intro's :

IRC Help <http://www.irchelp.org/>
IRC Prelude <http://www.irchelp.org/irchelp/new2irc.html>
IRC Intro <http://atpibm6000.tuwien.ac.at/~acvitkov/IRCIntro.html>
New IRC Users <http://www.newircusers.com/>
Cnet features
<http://www.cnet.com/Content/Features/Howto/Beyond2/ss01.html>
Get started w IRC <http://www.jetlink.net/~cyberguy/mirc/mirc.html>
Aide IRC en Francais <http://www.creaweb.net/irc/>

IRC FAQ's and Help :

IRC for normal people <http://wkweb5.cableinet.co.uk/DANDEB/ircmg.htm>
A Guide to IRC Netiquette <http://mirc.stealth.net/mircrulz>
Operators Guide <http://www.irchelp.org/irchelp/opguide.html>

Books about IRC :

Books about IRC <http://www.mirc.co.uk/books.html>
ChaTcIRcuiT: IRC Webzine <http://www.chatcircuit.com>
UnderCurrents <http://www.user-com.undernet.org/newsletter/>

IRC History :

The History of the IRC
<http://www.lick.pvt.k12.ca.us/~carmstro/networks/irchistory.html>
How IRC started (by Jarkko) <http://www.mirc.co.uk/help/jarkko.txt>

Undernet Interview with Jarkko <http://www.mirc.co.uk/help/jarkko2.txt>
Undernet History <http://www.undernet.org/documents/uhistory.html>
EFNet History <http://www.the-project.org/history.html>

Running your own IRC server :

wIRCSrv <http://www.wircsrv.com/>
ConferenceRoom <http://www.webmaster.com>
Microsoft Chat Server <http://backoffice.microsoft.com/product/chat/default.asp>

Technical IRC Documents :

IRC Proxy info <http://www.refresh.demon.co.uk/ircproxy.html>
IRC RFC <http://www.mirc.co.uk/help/rfc1459.txt>
CTCP Info <http://www.invlogic.com/irc/ctcp.html>
DCC Resume Protocol <http://www.mirc.co.uk/help/dccresum.txt>
mIRC's Color Coding <http://www.mirc.co.uk/help/color.txt>
EFnet Development <http://www.funet.fi/~irc/server>
IRC3 <http://www.the-project.org/>
IRC Server Numerics <http://www.teleport.com/~jeepster/numeric.html>

[Table of contents](#)

6 FEATURES, TIPS and ANSWERS to QUESTIONS.

6-1 How do I run the 32-bit mIRC on Win 3.1 or WFW 3.11 ?

The 32-bit mIRC no longer supports Windows 3.1 or WFW 3.11. Older versions used to work on Win 3.1 and WFW 3.11 if you installed the Win32s 32-bit extentions. There was and is no difference in functionality between the 16 and 32-bit mIRC apart from things old Windows versions simply dont understand.

On Windows 3.1 or WFW 3.11 simply use the 16-bit mIRC.

[Table of contents](#)

6-2 How do I run the 32-bit mIRC on Windows95, 98, 2000 or NT ?

On Windows95, Windows98, Windows 2000 and Windows NT no problems running mirc32.exe have to be expected. To use the 32-bit version of mIRC, you best install the standard 32-bit stack and dialer that are a part of the Windows 95, 98, 2000 and NT installation. The dialer then needs to be configured to access your Internet Service Provider. If you can use Internet already you have done this already! A standard install of Windows will work perfectly!

[Table of contents](#)

6-3 Is there a version for Windows CE, WebTV, Mac or Linux?

No, sorry. mIRC just comes in two tastes for Windows; a 16 bit and a 32 bit one. These issues will work on Windows 3.x, Windows95 and Windows98, but not on Windows CE. mIRC also does not work on the Atari, Commodore, WebTV, UNIX, Macintosh or Linux operating systems. We have no plans to make mIRC versions for these systems in the future either. Sorry....

Several people have reported to use mIRC succesfully on OS/2-Win and on Linux with the

Windows emulator Wine! I assume mIRC also works in a virtual Windows environment on the Macintosh and there is a perfect IRC client for Mac's, written by Onno R. Tjdgat, called Ircle (<http://www.ircle.com/>).

All IRC programs for WebTV, Linux, Mac's etc. are perfectly mIRC compatible. With all IRC programs you will see all your friends using IRC with mIRC and vice versa.

[Table of contents](#)

6-4 How do I solve the "Can't Resolve Host Name" error?

If you are using Windows95 or later versions, first find out if you're trying to run mIRC32.exe with a 16-bit winsock. You need a 32-bit winsock to run the mirc32.exe on Windows95/98. If you have a 16-bit winsock use the 16-bit mirc.exe. Also read section 6-2.

In the File/Setup/Local_Info dialog box, you'll see mIRC needs your PC's Local Host name and IP address number. This info is needed to be able to set up DCC Send/Chat connections to other users.

Normally spoken you could leave both fields blank, set 'On connect, always get Local Host and IP Address' to active with method 'Normal' and everything should work fine. But if you don't have an actual IP address (such as with TIA, Twinsock, SLiRP, or some other SLIP emulator), or if your stack is strange or there is some network oddity, it might not fill in these fields correctly.

The most simple solution is to switch to the alternate method and use the IRC server you want to connect to, to find your system's address information. Eventually clear the Local Host and IP Address fields first. Then set 'On connect, always get Local Host and IP Address' to active and select the 'Server' method. Now re-try to connect...

In the rare occasions the above method didn't help you could uncheck "On connect, always get IP Address and Local Host" and manually enter your PC's Local Host name and IP number. If you are using TIA, Twinsock, SLiRP, etc., simply enter your provider's local host name and IP, not your own. (Your provider's IP is not 192.0.2.1 or anything like that! That's a dummy IP used by these emulators. If you don't know your provider's local host name and IP, ask them!)

Also setting the Ident server to active might help. (See section 6-7) (Note: You must be disconnected from any IRC server before changing the Ident server settings.)

If none of the above solutions helped it's also possible that mIRC cannot complete the reverse look up it tries and gives the "Can't resolve host name" error when your IP name is not properly configured in your providers DNS. This is not something you can solve yourself. In this case you have to ask your provider to correctly assign an IP name to your IP Number to solve the problem. For your provider setting up the IP name on the DNS is not that hard, and doesn't take much time.

[Table of contents](#)

6-5 How do I solve the "Unable to resolve IRC server name" error ?

If you cant get mIRC connected to one particular IRC server and you get this error you should first give another server a try... If you can connect to any other server you should check if you didnt make a typo in the setup of your initial server address.

(File/Setup/IRC_servers/Edit)

If you get this error with whatever IRC server address you try, it is very likely your providers DNS (Domain Name Server) is down, malfunctioning or very slow. Especially if mIRC always worked flawlessly for you. Besides waiting or giving your providers helpdesk a phone call you cant get this fixed.

mIRC needs a DNS to translate (resolve) the IRC servers' IP Address you specified, into an IP Number. (Like the IP irc.law.emory.edu resolves to the IP Number 170.140.50.195) (To see this give the command "/dns irc.law.emory.com" in mIRC) You could bypass the need for the

DNS lookup by specifying the IP Number in your server setup instead of the IP Address for every server.

Under File/Setup/IRC_Servers/Add or Edit server, specify your IRC servers like :

Description : irc.law.emory.edu (Emory University)

IRC Server : 170.140.50.195

Port : 6667

If nothing helps you of course could also have a rotten or miss-configured winsock?

[Table of contents](#)

6-6 How do I solve the "Not enough user parameters" error?

You will get a "Not enough user parameters" from mIRC if you try to connect to a server but you didn't have the local host filled in, or had it filled in incorrectly. Check your entry under 'File/Setup/Local_Info/Local Host.' Also check to see if you supplied mIRC with a valid E-Mail address under 'File/Setup/IRC_Servers/E-Mail:'. (See section 8, too.)

Note: The e-mail address is used internally by mIRC and cannot be seen by others when they do a /whois on you. Some have noticed the "email:" field in the User Central dialog (/uwho command). This field is made up of your or the other person's user@host (your username plus your local host name)... in many cases you'll notice that it doesn't reflect the actual e-mail address. It has nothing to do with 'File/Setup/IRC_Servers/E-Mail:'. Enter the correct address there!!

[Table of contents](#)

6-7 How do I solve the "You haven't registered" error?

Although mIRC is shareware and you have to register it if you like mIRC and/or if you continue to use it after the 30 days evaluation period, this error message is NOT a message from mIRC. mIRC will not block features or cease to function if you haven't registered yet. This is an error message from the IRC network you try to use.

If you receive this message and/or you get disconnected that quickly, your Local Host name and/or IP number might be wrong, or not filled in at all. Look under 'File/Setup/Local_Info/', and check if the local host is correct and if the IP address (number) is filled in automatically... An easy way to solve a wrong IP Address is setting the 'On connect, always get' "IP Address" and "Local Host" to "ON" and restarting mIRC. If this doesn't help you could be trying to connect to a NON-public server!! Try another server to check this...

[Table of contents](#)

6-8 Why am I unidentified and what does it matter ?

An ident server normally is maintained by the Unix machine of your network provider... It is a kind of nameserver that guarantees your Identity. Since most standalone windows machines are not correctly backed up by an ident service, such a server is built into mIRC. IRC servers can do an Ident request to your Ident server and then expect a standardized kind of answer. More and more IRC servers require you to be identified in some way, and they will disconnect you if you're not identified ! Also if you don't react, or do so in the wrong way, they can decide to disconnect you.... You can check if you're properly identified by doing a /whois on yourself. The first line in the reply should NOT contain a ~ (tilde) or a - (minus). If you have a ~ or - in it try activating the Ident server.

mIRC's built in Ident server can be switched to active under File/Setup/Identd. Set it to :

User ID: <account name> (The part before the @ in your E-mail address normally)
System: UNIX (ALWAYS fill in UNIX !! not dos, win or *whatever* else !!)
Listen on port: 113 (The standard ident port number)

A problem that users behind a proxy or firewall will experience is that, despite checking the Ident Server to active, mIRC will never reply to an Ident query. This is because with these setups it never gets the ident request. The proxy or firewall won't pass through the Ident request from the IRC server to mIRC if you have not configured them (properly). That just means you'll be seen as nick!~account@machine.net which is not the end of the world, unless you happen to be using a server that requires an Ident reply and disconnects you. To fix this you have to make sure your Proxy passes requests on port 113 on to mIRC. If this seems impossible you will have to find another IRC server OR get your provider to set up an Ident service.

[Table of contents](#)

6-9 I can't get mIRC to DCC send or initiate a DCC chat! Getting files works fine..

In almost all cases that DCC sending and chatting mess up, the initiating (sending) party causes the troubles. In all cases where DCC Send or Chat stops functioning (suddenly) or never worked at all, this is caused by wrong Local Host and/or IP Address settings at the senders side. (The Local Host and IP Address settings are found in the File/Setup/Local_Info menu.) Typically in the case that your IP Address is not correctly set, the other party will get your dcc message and then try to confirm and connect to your incorrect IP Address. Of course, this doesn't work and you both sit there waiting. You will see the "Waiting for acknowledgement..." for ever while the other person -did- acknowledge correctly. You have a wrong setup, not the other person. This normally is a very simple to solve problem. Also if you cant get files from another person ask him/her to check the IP Address settings in his/her IRC client.

A good cure to solve these DCC problems is to start with this :

- 1) Disconnect from your IRC server.
- 2) Clear the current text in the Local Host and IP Address boxes under File/Setup in the Local_Info dialog.
- 3) Select both 'On connect, Always get' Local Host and IP Address options in the Local_Info dialog.
- 4) Set 'Method' to Normal.
- 5) Leave the Local_Info dialog and the File/Setup menu both with 'OK'.
- 6) Reconnect to your IRC server.

If this doesnt work experiment with the 'On connect' .. settings... !! ie. For instance, try to set your Local Host address by hand and let mIRC look for only the IP Address on each startup. If you still cant get it to work try to use the 'Server' method instead of the default Normal one. (Read what this does in section 6-3)!

Everybody..

Check your time-out settings! Make sure that your time-out values in DCC/Options are set large enough!! "Get/Chat Dialog time out after" and "Send/Get Transfer time out after" are recommended to be set to at least 60 and 120, respectively.

In case you never ever managed to get sending files or initiating a DCC Chat to work, not even after studying all these hints, it might be that your provider blocks these DCC connections by the kind of internet access they give you. Known to give problems are TIA, Twinsock, Slipknot and SLiRP. Also the use of a Firewall by your provider or a Proxy on your

local network will often block DCC connections!!

If you use TIA or Twinsock or Slipknot ...

If you use TIA (The Internet Adapter) or Twinsock, at this point you cannot use DCC send or initiate DCC chat (with any IRC client, not just mIRC). You may want to try SLiRP or vTCP. SLiRP was the first SLIP emulator to allow DCC sending and initiating DCC chat. (As you know DCC get should always work fine, whatever connection you have. Besides firewall blocking you that is.) Virtual TCP is tested and proofed to allow DCC sending and chatting. More info <http://blitzen.canberra.edu.au/slipr> and <http://www.infoexpress.com/vtcp.html>

If you use SLiRP ...

With SLiRP (currently running 0.95j) set File/Setup/Local_Info/ 'always get local host' on connect to Active, IP Address should be the fake IP used for SLiRP (10.0.2.15 usually). Then DCC Send, Chat, and everything else should work perfectly fine, even on Windows95/98 with the Dial-Up Networking. (thanks Dave!)

If you have Dynamic IP..

If you have dynamic IP (your IP address is different each time you log on), make sure that "On connect, always get:" in the File/Setup/Local_Info dialog is set to get the Local Host and IP Address. If these were already set to ON make sure the correct 'local host' name and 'IP Address' are found by mIRC... on some winsocks this is rather tricky... Also try the 'Server' method! If you have a non-compliant stack, mIRC may not be able to correctly find your local host (domain name) and IP. With dynamic IP addressing you are in trouble then !! DCC file sending and initiating a DCC Chat (contrary to file getting and accepting a DCC Chat) requires that mIRC knows your correct IP number. Even without an IP number at all, mIRC will work as far as normal chatting is concerned, but won't allow DCC file sending or initiating a DCC Chat.

If you have Static IP..

If you have a non-compliant stack, mIRC may not be able to correctly find your local host (domain name) and IP. In the File/Setup/Local_Info dialog, uncheck the options to "Always get the 'Local Host' and 'IP Address'" and manually enter your correct Local Host and IP.

A known Windows95 bug causes a lot of people to report that mIRC (and any other IRC program) gets/finds the old (now wrong) Local Host name and/or IP Address after switching Internet provider. This blocks their capability of DCC Sending files and Initiating DCC Chats. If, for some reason, no matter what you do, mIRC picks the user ID (Local Host name) from the Internet Service Provider that you no longer wish to use this is fixable by editing the registry. If you open Regedit and look at MyComputer /HKEY_LOCAL_MACHINE/System/CurrentControlSet/Services/VxD/MSTCP you will see the Domain and NameServer fields from your old provider. These fields will persist even if you uninstall Dial-Up Networking and re-install and go through the TCP/IP settings again ! The best way to solve the described problem is going to Start/Settings/Control_Panel/Network/ double click on TCPIP/ select DNS_Configuration/ and set the HOST field to the hostname (ID) you have on your new provider.

If you use Windows95/98 mIRC allows you to send long file names with spaces in them, but other IRC programs very often can not handle this. This might cause your transfers to fail. You might want to select mIRC's option to fill (up) spaces in such a long file name by an underscore.

Some people experience DCC File Send problems with mIRC on a windows system with Norton Desktop installed. mIRC then suddenly shuts down completely (sometimes with an

error message) as soon as you try to select a file to send. The problem is that Norton Desktop's feature called 'File Assist' conflicts with mIRC's DCC Send dialog. If you shut off File Assist entirely it will allow DCC transfers fine. Even just disabling the "3D look and feel" in the File Assist options menu helps already.

Another program known to give DCC Send problems is a software package called Long File Names by View software. It is something you might be running in the background and you might never think of it as the cause of your troubles. The problem is that when you use the DCC Send option in mIRC, the dialog that pops-up doesn't allow you to select files so you can't send anything. Selecting files is blocked by LFN and if you disable the LFN software all your DCC problems will be solved.

[Table of contents](#)

6-10 I can't get mIRC to DCC Get!

If you can't get files by DCC from somebody he/she might very well (almost certainly) have the problem described above in section 6-8. Especially if you can get files from others fine. Ask the sending party to check his IP Address settings.

If you get some error message like "invalid directory" or "cannot write to file" you have to check and fix the download directory settings in the DCC/Options/ dialog. Is the default set to a valid directory? It sounds stupid but also full harddisks block getting files very effectively !

NetCruiser, Twinsock, and Internet in a Box are not 100% Winsock compliant, and some users experience DCC problems with mIRC and other programs... complain to Netcom or the others... Some people also get the error "unable to create socket" on IBOX and other winsocks. If this happens, try to connect again (choose File/Connect or type '/server'). This sometimes works.

[Table of contents](#)

6-11 How do I use Copy and Paste?

In channel, query and DCC chat windows mIRC uses a nice select-and-copy-in-one-move feature. Copying text in those windows is done automatically as soon as you release the mouse button you used to select the text with. Just highlight the text and release. Isn't that a nice feature? Use ctrl-v to paste the copied text. The normal Windows' "Copy" menu is not just left out due to ignorance, but is due to the use of colored text in the graphical windows in mIRC. Windows does not support colored text in standard text boxes, and Khaled had to write all text box/windows routines himself! :-). And since using cut or paste in a channel, query or chat window makes no sense anyway, he was able to invent and implement this feature. All text shown in mIRC outside the channel windows (in settings, menu's, dialogs and on the edit box line) can be copied, cut and pasted normally using the ctrl-c, ctrl-x and ctrl-v key combinations.

[Table of contents](#)

6-12 How do I customize the colors in mIRC?

mIRC has a feature that allows you to fully modify the default color settings of all kinds of text and message windows. Look for the dialog under the Tools/Colours/ menu item. Virtually everything can be shown in whatever color you like most. By default the colors are set in a way that clearly distinguishes the various kinds of messages you'll encounter on IRC. Do not

hesitate to experiment; the Reset button in the menu will set all defaults back!

Besides the configurable colors you can set 'locally' in mIRC to indicate different kind of messages, you can also use colors on your text lines seen by others. Use the ctrl-K key combination to insert special control characters in your text. Add a number from 0 to 15 to use one of the 16 colors available.

[Table of contents](#)

6-13 How can I customize the fonts used in mIRC ?

All font settings can be done from the windows' System Menu's. (Those menu's under that little horizontal bar in every top left corner of ANY window in mIRC and windows. Access it by combining the ALT and - (minus) key) You can also use the /font command from any window.

Use /font or select the 'Font' option in the System menu and a fonts selection dialog will pop up. Here you can choose whatever fonts are available on your machine. This could be New Times Roman, Arial, MS Sans Serif,..... whatever you like. Also you can also set the font size and set it to bold. (italic is available but disabled for most fonts at the moment) The settings you set here will be saved as the default for the window you are in. To set a font as default for all windows of the same type, make sure to select the 'Use as default' checkmark!

[Table of contents](#)

6-14 How do I get mIRC to use 3-D dialog boxes?

Do not worry about this; recent mIRC versions automatically use 3D dialog interfaces. All you have to do is provide the proper ctl3d DLL(s). The dialog boxes will take on a nice gray color with a nice slightly "chiseled" appearance. The normal channel windows and or text wont be affected by this 3D look.... To see the 3-D dialog boxes in mIRC16, you first have to make sure that you have the file CTL3DV2.DLL correctly installed. CTL3DV2.DLL was not originally included in Windows or Windows for Workgroups 3.11, so you may not have a copy. Check your windowsdirectory. For proper installation, you should have only one copy of this file, the newest one, on your system, in the system directory. (Having multiple copies of the ctl3dv2.dll will mess up the behaviour of this dll !! Check your ENTIRE harddisk !!) You can FTP the file from ftp://bitsy.mit.edu/pub/dos/alpha-beta/techinfo/ctl3dv2.dll
In mIRC32 you need the CTL3D32.DLL. It has to be in the c:directory. Again make sure to have only ONE copy of this DLL on your entire Harddisk On windows95 make sure to have version 2.26 or later of this DLL !! Also WFW311 people can use this one. (You can see the version by using the EXPLORER, select the file, right click, select properties, version.)

[Table of contents](#)

6-15 What is a script?

A script is nothing but a combination of automated reactions your mIRC performs for you, often combined with simple commands to control or trigger these reactions. mIRC has three sections in the Tools/ menu in which it can be "programmed" in some way: The [Aliases](#), [Popups](#) and the [Remote](#) sections. With these sections, in combination with the Users and Variables sections, you can completely program and modify mIRC's behaviour on IRC. The combination of Aliases, Popups and lines in the Remote section usually is called a script.

[Table of contents](#)

6-16 How do I make a script?

First of all you have to make sure you really need a script. Scripting is not difficult but if you simply want to auto-op your friends on your IRC channel, or if you want to ban or ignore an annoying person, mIRC is perfectly equipped already with functions that do exactly what you want. In this case read the section about the [auto-op, protect and ignore lists](#).

In case you really need a script to make mIRC do some advanced tricks nobody thought of before, read study and learn the sections on [programming mIRC](#) in chapter 7 of this FAQ. I strongly discourage the use of scripts created by other persons. If you did not write your script yourself, chances are high you have no idea what exactly the script is doing and you, and your PC, are in the hands of the author of the script, a person who could very well turn out to be a malicious hacker. I'm not trying to get on your nerves here but you should realize scripts are very powerful. It is easy to write something that allows others to access and completely ruin your PC, or to read copy and spread all the private documents, email and passwords stored on your PC.

Note; By default mIRC does not come with any scripts or settings that can get you in trouble. The default settings in mIRC for DCC file transfers etc. are perfectly safe. All changes to settings that, in combination with other settings, or careless behaviour, could get you into trouble will display a warning message before any changes are made. Of course creating a script can be great fun. You can program lots of useful things in mIRC and I encourage you to share your tricks with your friends on IRC.

[Table of contents](#)

6-17 How do I specify people correctly in the auto-op, protect and ignore list?

When you look in the File/Options/IRC/Control/ menu in mIRC you will find the Auto-op, Protect and Ignore address lists with which you can easily auto-op and protect your friends on your IRC channel, or ignore an annoying person. This set of 3 lists provides you with the most basic channel maintenance possible in mIRC. No scripting is needed to use these functions. If you need better, smarter, but also more difficult to understand functions or behaviour, you should check out mIRC's 'remote' capabilities explained in section 7 of this FAQ. For starters these functions will do fine I think.

For mIRC to be able to op, protect or ignore a person on IRC it matches the nickname or address of that person with the nicknames and addresses you have in the op, protect and ignore lists. In the address definitions you can use wildcards.

The full address format of a person on IRC is Nick!Account@machine.

Lets assume that Bill responds to a "/whois bill" with:

```
Bill is gates@pentium.company.com * Billy A Lot Of Names Gates
Bill @#mIRC
Bill irc.server.company.com (A superb company)
Bill 666 seconds idle
Bill End of /WHOIS list.
```

If you don't care about channel takeovers and other annoying things, you can just specify the nick (Bill) of the person you want to auto-op, ignore or protect. If you want to be sure you 'op', ignore and protect the right person, and not a faker, specify them by their address if possible.

In that case, Bill would be best referred to by

bill!gates@pentium.company.com

You could also use *!gates@*.company.com if he works on several machines with different nick's....

##To add a user to the lists use the /auto, /ignore or /protect command (like /auto *!gates@*.company.com), or use the command while specifying an Address Type, or add a user by simply typing him into the specific edit box.
Make sure to switch the functions to active by the checkbox under File/Options/IRC/Control totally right of the function markers of the edit box. Or use the commands /auto [on|off], /ignore [on|off] or /protect [on|off] on the command line.

You can make the auto-op, ignore and protect more specific by specifying some parameters.

The auto-op and protect can be set to make people operator and/or protect them only on certain channels if you want. In that case just specify the channel names. like /auto <nickname|address> [#channel1,#channel2,...] and /protect <nickname> [#channel1,#channel2,...] (protect only works with nicks!)
If you dont specify one or more channel names mIRC will op and protect the person on all channels where you're operator.

The ignore can be set to ignore only certain actions from a person. mIRC distinguishes between private messages, text in channels, notices send to you, ctcp commands and invitations. From the command line you can use: /ignore [-pcntik] <nickname/address> [type] where p = private, c = channel, n = notice, t = ctcp, i = invite, k = color. (more parameters discussed in the help file!) In the ignore section an ignore will look like nick!userid@host.domain,private,channel,notice,ctcp,invite. If you dont specify any of these additional parameters mIRC will just ignore somebody totally.

[Table of contents](#)

6-18 What is the best script?

The best script is the script you wrote yourself.

mIRC's scripting language is very flexible and easy to learn. By reading the Help file and the sections in the end of this mIRC FAQ, you easily learn the basics of mIRC's scripting. More advanced scripting questions can be asked to the usenet newsgroup alt.irc.mirc. If you don't know what to write in your script, or if you have noo idea how to start, try looking in some of the pre-fab scripts spread over the web and on IRC.

As always beware of using a script you do not fully understand. Many scripts writers included backdoors in their scripts that can make you do things that are out of your control. Therefore read reviews and documentation from and about the script.

Many help channels refuse to help with pre-fab, or third party scripts. The only way to get help with such a script is from the author or other users of the script. Most mIRC related help channels on IRC -will- help you with stand-alone remote lines and small scripting problems.

[Table of contents](#)

6-19 What are Trojans? What are script.ini and dmsetup.exe?

In short; Trojan horse attacks are attractively disguised files that you download and run, resulting in harmful and dangerous consequences ranging from takeover of your IRC channels, erasing of your hard disk, theft of your account passwords, etc. These (Trojan)

virusses are not mIRC or IRC specific, they just spread like fire on IRC.

Trojans are typically files with suffices like ".ini", ".exe", or ".com", such as "dmsetup.exe" or "script.ini". These days nearly all trojans are spread in the guise of a free game, handy tool or other software. You probably downloaded one from a WWW or FTP archive, ICQ file exchange, or through IRC's DCC file transfer (by manual /dcc get or, even worse, an "auto DCC get" feature which allows anybody to send you anything, including not only trojans but also other viruses, child porn, etc).

Typically the Trojan needs to be run manually at first (by you), and then installs hacked files all over your disk silently. There are many different versions of those files, but almost all of them interfere with your mIRC placing backdoors in scripts. The files then auto-send themselves (using an 'ON JOIN' event) to everyone who joins the same channel as an infected user without the users knowledge.

At <http://www.irchelp.org/> you will find detailed instructions and information on all kinds of problems you may encounter on IRC. At this site the best help for problems like this is concentrated and organised by people who are on IRC 24/7, in the Help channels and alike. Read <http://www.irchelp.org/irchelp/security/trojan.html> to learn all about the virusses on IRC, mostly called Worms or Trojans, that might tackle you.

Prevention: NEVER download files from people or sites which you aren't 100% sure about. Never use the "auto DCC get" feature, and always scan your DCC gets with a decent virus scanner. Note that mIRC by default does NOT accept files from strangers. This has never been otherwise either. If you accepted files by the "auto DCC get" feature in mIRC, you have switched this option ON yourself, really. Do not, never ever, accept anything you have not requested. Do not accept anything from someone you don't know, no matter how attractively packaged.

Removal: Removal of script.ini, dmsetup.exe, and other trojans is a difficult subject. The many variations of the files have different removal techniques. <http://www.irchelp.org> is a great information resource for removing these worms. Research all information resources before trying to remove the worms to help determine the best removal techniques.

[Table of contents](#)

6-20 How can I make mIRC react to Remote Commands?

Use the Tools/Remote/ section.....

mIRC can react to CTCP commands given by other users. You can customize your responses to CTCP commands and offer files: features that usually require scripting but here are handled by the Tools/Remote/Commands section. It is this section in mIRC that can handle OPME, KICK, MYLEVEL, or XDCC SEND commands for you, if you configure it well. Setting up the remote section is not an easy task, but the way it works guarantees full freedom to make it do what you want it to do.

As soon as your mIRC receives a CTCP command from somebody else, it checks if that command is defined and if so, it matches the required user level against the level of the remote user. If the remote user has a suitable user level his command is executed... All commands given to you have to have the format '/CTCP <yournick> <command> <parameter(s)>'. If a command is defined on several levels, the highest ranked one is executed. See the help file included in the mIRC39.zip package for detailed instructions and read the tutorial in section 7 of this FAQ.

[Table of contents](#)

6-21: How can I make mIRC react to Events?

Use the Tools/Remote/ section..... Events are all things happening on channels or in private conversations on IRC. People joining, leaving, getting opped, deopped, greeting you, even plain talking all are Events... You can configure mIRC to react however you like to almost anything that can happen in IRC. See the mIRC help file for further instructions and read the tutorial in section 7 at the end of this FAQ.

[Table of contents](#)

6-22 How do I specify my friend Bill correctly in the Users list with an access level of 2?

(In the Tools/Remote/Users section..)

You can add your friend Bill with access level 2 to the Users list in two ways:

1. Specifying the nick -- "2:bill"
and/or by..
2. Specifying the address -- "2:bill@gates@pentium.company.com"

Of course, specifying the address is the safer option. Wildcards are supported, such as "2:*!gates@pentium.company.com". In which case all nicks bill could use on his account are recognized. To get the ON OP, ON DEOP, ON SERVEROP and ON NOTIFY events working you *have* to specify a person by his nick !! It won't work if you specify his full address only... In the case of bill, you could use BOTH Users list lines mentioned above. You could just type the nick and/or address straight into the Users section under Tools/Remote but mIRC also has special commands to add people to your Users list straight from the command line. Look in the mIRC help for the commands /user, /guser and /ruser.

[Table of contents](#)

6-23 I get disconnected after the /LIST command.

It is quite common that everytime you do a /list, you get disconnected part way through. Even if you try a partial list, eg. "/list #love", it runs for about 90 seconds and then disconnects you. Sometimes using a local IRCserver helps.

You didn't do anything wrong... It is important that you realize what happens if you do a /list command. The server generates a list of all channels (about 2000 on Efnet) and quickly sends that list to you. BUT, because the data throughput isn't infinite, the whole output of the /LIST command is queued in a buffer on the server. At some point that buffer gets overrun, and the server, detecting this ('reached maxsendq'), disconnects you. This mechanism is made to disconnect people who generate more characters per second than any 'normal' person uses for normal conversations. It's a protection mechanism, but unfortunately the server does not recognize that the data stream you caused is simply the result of your harmless /list command, and disconnects you.

You can also tell mIRC to show only channels with a minimum and a maximum number of people. Also, if you specify a #string, then mIRC will only list channels with that string in their title. BUT, for the problem described, this does not matter/help at all!! The server ALWAYS returns the entire #channel list and mIRC (like all other clients) takes care of the filtering!! So, asking for a partial list to prevent you from disconnecting won't make ANY difference!!

So, this isn't an mIRC bug. It's a pity, but there's nothing you can do to prevent this from happening. It's just one of all-too-many IRC oddities. Try using some other servers, as many aren't quite so touchy. To facilitate users on relatively slow modem connections the list of channels is also saved to a file "channels.txt" in the mIRC directory. The channels list dialog has "Apply" and "Get List" buttons. If "Get List" is pressed then a fresh list is read from the server, if "Apply" is pressed then the latest list you downloaded earlier is used to search for channels etc. This means that you once you successfully retrieved a list once, you can decide to use that list in later IRC sessions! Or you can share lists with other people. Of course a list won't be totally up to date everytime but for the major channels that doesn't really matter and you'll never get list-disconnected again :-)

Oh ... check out the list window popup menu !! Once you've done a "/list" the list is stored in memory and available for re-processing with keys, min/max settings and additional filtering! Right click in the channels list window and play with the parameters you can set!

[Table of contents](#)

6-24 How can I protect myself against flooders?

You can protect yourself against people who are flooding you with the new automatic anti-flood system. Look in the File/Options/IRC/Flood/ section. A server usually disconnects you for sending too much data to it in a certain period of time, *or* if you try to send it data when it hasn't finished processing your previous data. The new flood protect makes sure - others- can't make you to send too much data to the server.

Usually a server has a buffer of about 512 bytes. mIRC therefore counts the number of bytes you've sent to a server and if this exceeds a certain number, mIRC waits for the server to be ready again, before it continues sending data. In the mean time it nicely buffers unsend lines. This should protect you properly from all sorts of ctcp floods and so on... You set the amount of bytes mIRC may safely send (for instance 350 bytes), the amount of lines it may buffer (like 20), the amount of lines it may store maximal per user (like 3) and how long the flooder should be ignored (like 30 secs) by the command /flood 350 20 3 30 This flood control method *only* works for messages being triggered by other users. So you can still flood *yourself* off the server. (like with the /list command)

[Table of contents](#)

6-25 Running multiple copies of mIRC.

mirc.exe 16-bit :

You can't just start the 16-bit mirc.exe a second time. Windows prevents this for 16 bit programs and besides this, all your settings in the mirc.ini files might get messed up a little, if two programs access them at the same time! If you want to run a second copy of mIRC16 (don't ask me why), it's best to install mIRC a second time on your harddisk, in a new directory like c:\mirc2\. This prevents all possible things you could mess up easily ;o) The mIRC installer program will nicely create a new start icon for you, just give it a proper name, like mIRCBot. You can now start both exe's and configure them independently..... Nice huh?

mirc32.exe 32-bit :

While the 32-bit version of mIRC could easily be started a second time, I would still advise you to just install a complete new instance in a new directory. This takes about half a meg disk space extra on your harddisk but also makes sure it will work 'out of the box'.

[Table of contents](#)

6-26 How do I make myself invisible to other users ?

And what it does and doesn't do. ;o)

In mIRC, you can make yourself invisible with: `"/mode <your_nickname> +i"`. The title of the STATUS window will show (+i) after your current nickname. If you change nick, you will remain invisible under your new nick. Use `"/mode <your_nickname> -i"` to make yourself visible again.

##You can also set the 'Invisible mode' option in the File/Options/Connect/ menu to active.

Nobody can detect your changing from invisible to visible or vice versa. When you're invisible, you are always visible to all the people who are on the same channel(s) as you are. When you join a new channel (while invisible), everyone in the channel sees you joining. Also, leaves are NOT hidden. Also, when people do a `"/whois <yournick>"` they will see a normal whois list on you if the nick they enter matches your's exactly.

So, what's the point of this "invisibility" capability ?

Users NOT on the same channel as you WILL NOT be able to see your name if they do `/who <#channelname>`. Also, if they do `/names` to list all the people currently on IRC, they won't see you. Also doing `"/who *part.of.your.address.net*" won't result in your nick turning up. The point of being invisible is not to be able to stroll along channels like a ghost without anyone seeing you, but to hide (a bit) from users that scan channels with /who to find you. If they scan channels by actually joining them, then invisibility won't help you. To get rid of annoying people: first make yourself invisible ... then change your nick and make sure those people can't join the channels you are on in order to find out your new nick. (also, doing: /ignore <other_persons_nick> helps a lot !`

[Table of contents](#)

6-27 How does the sound command in mIRC work ?

Assuming you have a sound card or a speaker driver allowing you to play sound files on your PC, you can make some funny use of sound on IRC with mIRC. Already available in mIRC before version 3.7 was the `/wavplay <c:\path\sound.wav>` command which allowed you to play a .wav sound file locally to you. In version 4.7 playing of Midi files is added and the command renamed `/splay`. With this command you can for instance play wav's and midi files to alert you when your friends join IRC or when people get kicked or whatever else. You can 'program' these events in mIRC's remote section.

The `/sound` command allows you to send a request to any other party to play a .wav or .midi file he and you both have. First set mIRC to Accept sound requests under File/Options/Sound Requests/ The command syntax is `/sound [nickname|#channel] <filename.ext> [action text]`. As you see in the play request an action text may be specified which will display on the other side. I have to stress that the actual wav or midi file is NOT sent to the other party. We dont want to stuff the net with data you know. The command just triggers playing of the sound file at your and somebody elses machine. So make sure that the other party has the file you want to start. A `"/sound friend tada.wav does a tada"` command done by you will result in `'*friend* does a tada'` and the tada sound at your side (the *friend* confirms the destination to you) and the action `"* afriend does a tada"` with the tada sound played at your friends side...

This command currently only works in between mIRC users and people using compatible clients! The command format is `/sound [nick|#channel] <file.ext> [message]`. The message

will show as an action to the receivers and both you and they will hear the wav you selected. mIRC will look for your sound files in the directory you set under File/Options/Sound Requests/

Playing with sound you might like the /speak command too! This command makes mIRC use the Text To Speech program Monologue (by DDE) to speak up whatever text you add to the command Read more in section 6-31.

[Table of contents](#)

6-28 How does the WWW support in mIRC work ?

mIRC supports the Netscape, Mosaic and MS Internet explorer WWW browsers. You can now surf the www with your friends and exchange addresses of the best pages you find. mIRC's URL catcher automatically stores URL's from text passing by in channels, privates and topics in a listbox. You can also send one or more URL's to the channels you are on (or to any private conversation you are in) by the URL window popup menu. (Popup menus are activated by the right mouse button !!) You can store URL's for future reference and set them in any order you want by the use of markers. Newly found items are appended to the end of your URL list with the '?' marker. When the marker for a URL is changed to something else it is sorted into the permanent list. In the URL options dialog box you can set mIRC to delete '?' marked items on exit. In the URL window you can use the delete key to delete marked URLs and the Insert key to add a new url. Also a new string is created which represents the page your Netscape currently shows. You can use this \$url in popup menu definitions and in remote definitions.

To get the hang of it first enable the URL catcher in the File/Options/URL_Catcher/ menu. Also fill in the place where your www browser can be found, if mIRC didnt find it already. From that moment on mIRC will find and store all www addresses passing by on the channels you are in. You can see this going on if you open the URL window (use the URL button on the Toolbar or the /url command). Then if you want to view an URL in your www browser, just highlight it in the URL window and select 'view' from the URL window popup menu (right mouse button). Alternatively you can set mIRC to react to a double click on an URL in the URL window. Once you start to use the URL catcher you'll get the grip of it... Have fun !

New in mIRC is the possibility to join channels on IRC by clicking on a link on the www. You just have to configure the mIRCLink helper application into your Web browser. To learn more read the info pages on the mIRC www pages.

[Table of contents](#)

6-29 What is the built in finger server for ??

Lets first explain that there are 2 sorts of fingering on IRC. First there is the IRC finger that almost all clients can handle. This is a CTCP command with the syntax '/ctcp nickname finger'. The reply you will get to this CTCP command is usually a one liner set by the user you do the finger on. In mIRC you can set this reply under File/Options/Action_Lists/ at the Ctcp_finger_reply line. Some other clients use the alias '/finger nickname' to shortcut this CTCP finger command but thats not the official way..

The other finger command comes from the finger system familiar to UNIX users. On UNIX systems you can finger the address of another user to find out some more about that person. Normally this fingering will deliver you the persons home address, telephone number or other usefull info. This info is given to you by a central finger server and you need a finger client to get it. mIRC already has a finger -client- built in for a long time under

Tools/Finger. If you specify a users address, mIRC will try to get some info on the person at the other end. The '/finger <nickname|users_address>' command is doing the same thing. (If you specify a nickname mIRC will try to find out that users address and finger it.) Mind the syntax difference from the IRC finger command !!

In the newest mIRC, a finger server is built in to enable you to provide finger info even if your school, provider or company doesn't maintain a central finger server. Under Misc/Options/Servers/ you can enable this Finger server. You also have to provide a text file with the info you want to give. This file should meet a certain syntax as explained in mIRC's help file. A last remark; dont be surprised if you find that the UNIX finger is hardly used on IRC !

[Table of contents](#)

6-30 How can I easily serve my collection of shareware to others, using mIRC ?

As a lot of people on IRC promote and share the best they found around on the net, mIRC now offers a unique built-in Fileserver. This Fileserver feature is somewhat of a cross between DCC and FTP. You open the server window to someone, (it's a special DCC chat window), restricting them to a certain directory tree, and they can browse your file listings, change directories, read text files, or get files.

The syntax to set up a DCC server connection to somebody is:

```
/fserve <nick> <max gets simultaneously allowed> <homedir> [welcome file]
```

"Max gets" is so that the other person doesn't bring down your machine with too many parallel gets. 4 is probably a reasonable number. The other person will have access to his homedir and all dirs DOWN in the directory tree from that homedir on. "Welcome file" is a text file you can write and specify that will welcome users to your file server. It's optional. Examples:

```
/fserve Krejt 3 c:\temp\serve c:\temp\servng\welcome.txt  
/fserve Mookies 2 c:\outgoing c:\network\mirc\welcome.txt  
/fserve Friend 7 c:\
```

Keep in mind that you can't set up a server to yourself... you need others to test your server...

Typing help in the file server will show the available commands, which are styled after Unix and DOS. "ls" or "dir" will show a directory listing, for example. Even switched commands like "ls -k" (show file sizes in kilobytes) and "dir /w" (show a wide directory listing) work. The server supports all normal ftp commands like cd <dir> , cd.., dir, ls, get, but NOT put, hash, upload etc. There is no possibility to delete files in a server connection. Safety risks are none or minimal due to the major restricting of available commands.

Of course, the /fserve command can be used in your Remote section....

Set up a simple Tools/Remote/command like :

```
1:server:/fserve $nick 3 c:\temp\serve
```

Set the commands to active (/remote on) and off you go....

Other people only have to type "/ctcp yournick server" to activate the server. You can't set up a server to your own mIRC!! So, others have to test your server !! In the directory c:\temp\serve, you place all files other people are allowed to get from you. The people using your server will have access to the c:\temp\serve directory AND ALL directories BELOW it.. like c:\temp\serve\games.

[Table of contents](#)

6-31 How do you use the Text To Speech support ??

In the new 32 bit mIRC 5.7 the Microsoft Agent software is supported, through scripting, if you have it installed on your system. An agent is an animated character that can speak text and perform actions. With a little fiddling and tweaking you will be able to have this agent speak to you, warn you about things happening on IRC, or even read-up entire discussions to you! You have to download and install some files from the Microsoft website. Read more on <http://www.mirc.co.uk/agents.html>.

For older mIRC versions you can use the software packages 'Monologue' or 'Text Assist' to speak out text at your wish. You can make it to say out loud whatever you want that happens on IRC; on channels or in private chats etc.

Monologue was sold in a software package with older Soundblaster sound cards. Monologue is not free and not shareware either ! It is made by the Company First Byte (<http://www.firstbyte.davd.com/html/fbinfo.htm>). The software is some years old but the 16 bit version works rather well.. and a lot of people have it. Therefore mIRC supports it to have Text To Speech support. I heard a 32 bit version has released as well but mIRC is NOT tested to work with this version and I fear it doesnt even support the needed DDE routines. First you have to make sure Monologue is properly installed on your PC and working in good order. It has a built in test function and should work properly with the test and clipboard support before trying it from mIRC. To use it with mIRC you have to enable monologue's DDE support. Through DDE mIRC communicates with it. Once you enabled the DDE support you can fire up mIRC and give it a try.

Text Assist is the newer Text to Speech application included in the Soundblaster sound card packages. To use it together with mIRC you first have to make sure you have the tassdde.exe helper application. Get it from the Creative Labs support pages at <http://www.creaf.com/creative/devinfo/tassdde.exe> To use Text Assist now just run this helper application first (and next to mIRC) and then in mIRC use the /speak command.

In mIRC the '/speak <text>' command is used to send text to Monologue. It is used like :

```
"/speak this is a test"  
"/speak can you hear me ?"  
"/speak this is fun"
```

It is fun indeed ! Once you manage to speak one-liners from a mIRC window by this /speak command, it is not that hard to get it speaking whole channels or conversations ... assuming you can handle programming mIRC's remote events section. (Read and study section 7.) It is not meant to be a plug and play solution ;o) mIRC's support for Monologue and Text Assist is realized through its built in DDE engine. To learn more about the use of DDE to interact with other programs running on your machine read the mIRC helpfile.

[Table of contents](#)

6-32 What is a Firewall or Proxy? How can I use it?

-- A firewall is not a feature, it is something you need to solve, something to get around. -- One potential drawback to an Internet connection is that other Internet users could gain access to files and data residing on your PC. This can be prevented by establishing a security system such as a "firewall". A firewall usually consists of a UNIX machine that is placed between your PC and the Internet, and all traffic between you and the Internet must pass through that machine. So a firewall is not something included in mIRC, or something active on your PC. The problem is that most firewalls also block your possibilities to use IRC.

Normally spoken firewalls block all IRC traffic and access to an IRC server, unless specific precautions are made and a Socks compliant firewall is used.

If your provider or employer uses a Socks compliant firewall that blocks your IRC access you can set mIRC to connect to the firewall to reach the outside world. In this case, and -only- in this case, set mIRC to use the Socks firewall under File/Options/Connect/Firewall/. Your provider or employer should be able to provide you with the necessary settings.

In other, similar, situations you can use a HTTP or HTTPS Proxy to connect to IRC with mIRC. A HTTP Proxy is NOT something like Internet Connection Sharing, WinProxy or WinRoute! For HTTP Proxies in the /File/Options/Connect/Firewall/ menu in mIRC select the 'Use Firewall' and 'Proxy' options. Fill in the HTTP Proxy Hostname and Port settings. Normally you can copy these from your webbrowser! For more info see <http://www.mirc.co.uk/help/proxies.html>

If you never heard about firewalls before it is VERY unlikely you are behind one! In this case stay away from the firewall settings and dont enable it. Also if mIRC works for chatting but things like DCC File Sending are blocked this normally can not be due to a firewall problem, and enabling mIRC to use a Socks firewall or HTTP Proxy will not help you!

[Table of contents](#)

6-33 How do I use mIRC's built in DDE support ?

This section on mIRC's built in DDE support is initiated by Prophet's quick DDE Reference. Thanks for allowing me to include parts of your work in the mIRC FAQ!

What is DDE? - DDE stands for Dynamic Data Exchange. DDE is used by programs running under Window to communicate with each other. In the typical case one of the programs involved acts as a DDE server, the other as a DDE client. The server is always accepting requests and commands and the client is requesting information and giving commands to the server. Good examples of the use of DDE are mIRC's support of WWW browsers by the build in URL catcher and the support of the text to speech program 'monologue' by the /speak command. Read more about them in the sections 6-22 and 6-25.

Setting up mIRC's DDE server - To be able to control mIRC from within another application, program or another copy of mIRC running on your PC, you first have to make sure the DDE server is set to active. The server settings are found under File/Options/Servers/. You should always specify an unique service name but for now the default 'mirc' will do fine. The service name is used by mIRC to identify DDE commands given to it. To be able to play a little with the DDE things having only one copy of mIRC running will do fine for now, but for usefull future use you might have to start a second copy of mIRC or get another external DDE client/server. If you want to have a second copy of mIRC running dont forget to give it an unique DDE service name! The default server name for mIRC is 'mirc', but you can change the server name to anything more suitable.

How is DDE used? - To use DDE to provide information (or give a command); a POKE, or to ask for information; a REQUEST, you just need two simple commands.

The /dde [-r] <service name> <topic> <item> [data] command. The /dde command defaults to sending a poke unless you specify the -r switch in which case a request is sent. If you are sending a poke then all four arguments including the 'data' field are mandatory, if you are sending a request then the first three arguments are mandatory. This is why you might have to use a "" in the 'item' field; it acts as a filler and isn't actually used for anything. This filler is important because you might have the /dde or \$dde mixed with other commands or text in an alias, and mIRC has to

know the exact number of parameters when parsing. At the moment only one poke 'topic' exists; the COMMAND.

Instead of using the /dde -r to send a DDE request its often a lot better to use the \$dde identifier. The \$dde <service name> <topic> <item> identifier sends a request and retrieves any data returned by the specified service. All three arguments are mandatory (as explained in the above paragraph). Any retrieved data is inserted in the position in the alias or command where the \$dde was issued. Several request 'topics' exist; CHANNELS, CONNECTED, EXENAME, INIFILE, USERS, SERVER, NICKNAME, VERSION, PORT.

COMMAND topic. (Only for POKE's)

This allows mIRC to control a second client similar to the remote CTCP DO command. The command topic is used in conjunction with the /dde command like /dde <service name> command "" [instructions]

Here are a few examples, assuming your mIRC uses the default DDE service name, 'mirc':

```
/dde mirc command "" /join #test123
/dde mirc command "" /msg #test123 hello everyone
/dde mirc command "" /notice $me testing 1 2 3
```

The /dde command can be used in the Remote/Events and Commands Section in the Alias section and straight on the edit box.

CHANNELS topic (Only for REQUEST's)

The CHANNELS topic is a REQUEST which returns a single line of text containing the current channels the listening mIRC is on. The requests can be used from the command line, an alias, or in a remote. To be able to use and process the returned data mIRC uses the identifier \$DDE to represent the data. Note that the active channel is marked by a star!

Examples:

```
/say $me is currently on $dde mirc channels ""
/echo 6 $active I am currently on $dde mirc channels ""
```

CONNECTED topic (Only for REQUEST's)

This topic returns a single line of text with "connected", "connecting" or "not connected".

Examples:

```
/echo 6 mIRC is currently $dde mirc connected "" to a server
/echo 6 mIRC is currently $dde mirc connected "" to $dde mirc server ""
```

The other topics for REQUEST's are EXENAME, INIFILE, USERS, SERVER, NICKNAME, VERSION and PORT. Lets just give some examples:

```
/msg $active The mIRC exe name path and filename is $dde mirc exename ""
/echo 6 mIRC's ini file is $dde mirc inifile ""
/say users on #mirc are $dde mirc users #mirc
/echo 1 (would) use $dde mirc server ""
/echo 3 my current nickname is $dde mirc nickname ""
/echo 4 My version is $dde mirc version ""
/echo 5 mIRC is connected to port number $dde mirc port ""
```

[Table of contents](#)

6-34 I switched to a new provider but mIRC keeps using the old one !

A known Windows95 flaw causes some people to report that mIRC (and several other Internet programs) tries to use the winsock software of an old provider, even if all references to that provider seem to be deleted from Windows95's setup. This blocks the capability of

DCC Sending files and Initiating DCC Chats. If, for some reason, no matter what you do, mIRC picks up the software, user ID (Local Host name) from the Internet Service Provider that you no longer wish to use, you can fix this by editing the registry. If you open Regedit and look at MyComputer /HKEY_LOCAL_MACHINE /System /CurrentControlSet /Services /VxD /MSTCP you will see the Domain and NameServer fields from your old provider and maybe even references to old winsock software. These fields will persist even if you uninstall Dial-Up Networking or other winsock's and re-install and go through the TCP/IP settings again ! The best way to solve the described problem is going to Start/Settings/Control_Panel/Network/ double click on TCPIP/ select DNS_Configuration/ and set the HOST field to the hostname (ID) you have on your new provider. If this fails edit the registry by hand, but do this with care!!

[Table of contents](#)

7 SOME NOTES ON 'PROGRAMMING' mIRC (Tutorial?)

mIRC has three tool sections in which it can be "programmed" in some way: The Aliases, Popups and the Remote section. In the Remote section you can define Users, Variables and Scripts. In Scripts you can define how mIRC reacts to things happening on IRC, in CTCP's and Events.

The best way to learn and understand the way these sections work is by studying them in consecutive order. Really! Please don't try to take a shortcut to the Remote sections before you really understand the Aliases and Popups. The lessons you'll learn from each are cumulative.

(A note for those familiar with ircii: mIRC's aliasing syntax is similar to ircii's, but not exactly the same. mIRC's Remote Command and Event sections allow you to implement many of the same things you can do with ircii scripting, but the structure and syntax are a bit different.)

[Table of contents](#)

7-1 The Aliases section.

Select the menu item Tools/Aliases/.
The mIRC Editor will start with the Aliases section opened.
Please ignore the Popups, Remote, Users and Variables sections for now!?

The mIRC Editor is used as a compact and uniform interface to almost all mIRC's programmable sections. With the Aliases section selected the 'File' menu will help you to load, unload and save alias files. The 'Edit' menu offers copy/paste and search routines. The 'View' menu lets you select what alias file you want to edit and 'Alias' shows an index of the current opened alias file.

In the Aliases section you can define shortcuts for often-used commands such as join, part, quit, opping and deopping, and whatever else you want. Even things you frequently say can be placed in an alias... In general, ALL commands which can be used in the edit box (the line or lines you normally use to talk/type in) can be used in an alias. Aliases can also be used to execute several commands at once. You can even define aliases that are launched by the function keys F1 - F12.... (Finally some use for those never touched keys..) In mIRC you can even call aliases in aliases or from popup menu's and remote lines. In the new mIRC you can have multiple alias files. mIRC comes with one prefab alias.ini file you can add items to it or you can decide to add an entire file with aliases. Very easy!

7-1-1 Creating aliases.

OK, let's start right off the bat with an example. To join a channel, you normally have to type:

```
/join #<name>
```

...where #<name> is the name of the channel, of course. Since this command is used so often, it'd be nice to use it by typing as little as possible, right? So let's create an alias for "/join #<name>" -- let's create a shortcut.

Instead of typing "/join #<name>", let's make it so that we can type "/j #<name>" with the same results. In that case, the Alias would be:

```
/j /join $1
```

(You don't have to understand exactly what that means to mIRC at this point, just what it will do. I'll explain the alias syntax shortly.) Give it a try by placing the line "/j /join \$1" in the Aliases list under Tools/Aliases (if it isn't already there). Then try it out on the command line (where you normally type) with a simple command like:

```
"/j #test".
```

You'll see that "/j #test" now does exactly the same thing as "/join #test" -- it joins you to a channel named #test. This construction is called an alias.

Alias Syntax.

All aliases use the format /<aliasname> /<command>

In other words, first your "shortcut" name (like "/j" in the example above), then a space, and then the actual command your shortcut is pointing to (like "/join" in the example above).

In alias definitions, you do not -have- to specify the / command character for commands since it is really only needed when typing commands into the editbox. Just for clarity I decided to use the / in this tutorial. Keep in mind that the alias "j join \$1" will work as well.

Aliases also use a number of special strings and identifiers. There are several strings such as \$1, \$3-5, \$\$1, \$?, \$!, # \$1, # \$ \$1, and even # \$?. Strings are filled in by whatever is specified when the alias is called. They are very dynamic in their use. Identifiers are similar to strings but are filled in by a standard 'value'. Important Identifiers are the #, \$me and \$url but the list of identifiers is long and ever growing. To give you a better idea; \$read \$date \$adate \$fulldate \$time and \$day are some other identifiers. Don't worry, we'll explain all of these in good time.

Right now, let's concentrate on \$1. After all, that's the string we used in our example above:

```
/j /join $1
```

\$1 means "the first word typed after the Alias, whatever that may be." So if you type "/j #test", mIRC knows that this is a shortcut for "/join #test". If you type "/j #hello", it will know that this is a shortcut for "/join #hello".

Multiple Commands.

Now, let's say that you'd like to join several channels at once. You can make an alias to do this. Just separate the commands with the | character (on most keyboards, this is the shifted version of the backslash key)... Let's say that you'd like to be able to type "/j3" and

then follow it with the names of three channels, and they'll all be joined at once. The alias would be:

```
/j3 /join $1 | /join $2 | /join $3
```

So if you placed this alias in Tools/Aliases and you were to type:

```
/j3 #test1 #test2 #test3
```

...mIRC would "/join #test1", then "/join #test2", and then "/join #test3".

In this case, you see that \$1 means the first word typed after the Alias. And \$2 of course means the second word, and \$3 the third. Now, mIRC is obviously expecting three words -- your channel names. So what would happen if you just typed "/j3 #test1"? mIRC would complain about the missing parameters, as it's expecting more channels to be named. There is a solution to this problem, the \$\$1 string, which will be explained later.)

Example: /say

Let's say that there's something you find yourself typing into a channel all the time, like your home page address, a greeting, etc. Take a look at these alias examples using the "/say" command: "/say" simply types the indicated text into your current channel or private message window.

```
/hello /say Hello to all of you...
```

```
/www /say The mIRC page is at http://metro.turnpike.net/mookies/mirc.html
```

```
/lines /say This is the first line | /say This is the second line | /say This is the last line
```

The last examples show how to /say multiple lines of text in one alias. Just separate the commands with the | character. You also see that not all aliases always have Strings in them.

Example: /me

Actions in an alias...

```
/fun /me thinks this is fun!
```

```
/hug /me hugs $1 firmly!
```

The second example uses a string, \$1. If you were to type "/hug everybody", this would be equivalent to "/me hugs everybody firmly!"

Example: Shared Strings

Multiple commands can also share the same string.

```
/jtest /join $1 | /topic $1 just testing
```

In this alias definition the parameter \$1 is used twice. If you were to type "/jtest #test", it would first join the channel #test, and then change the channel topic to "just testing."

Another example along the same lines:

```
/love /say I love $1 ! | /say I really love $1 !!
```

If you were to type "/love everybody", mIRC would know to "/say I love everybody !" and then, "/say I really love everybody !!"

Example: Mode Changes (Opping, etc.)

To put your often-used mode changes in the aliases section, you first need to know the normal structure of the mode commands. Opping somebody on IRC normally requires the command:

```
/mode #<channelname> +ooo <nick1> <nick2> <nick3>
```

...in which up to 3 persons can be mentioned.

This can be nicely aliased with:

```
/op /mode # +ooo $1 $2 $3
```

To op John, Jake, and Joan, now you can just type: "/op John Jake Joan"

The # in this alias shows the use of one of other the pre-defined strings (# and \$!). The # always stands for the #channelname of the channel on which the alias is used....

In the same way you can make aliases like...

```
/deop /mode # -ooo $1 $2 $3
```

```
/ban /mode # +b $1
```

```
/kick /kick # $1
```

```
/dbk /mode # -o $1 | /kick # $1 | /mode # +b $1
```

IMPORTANT: Proper spacing!

This is very important! You MUST include spaces between strings and the rest of your text! For example:

```
/like /say I really like $1!!!
```

...will NOT work! There needs to be a space between the string "\$1" and your "!!!"s. (This can sometimes make your text look awkward when using commands like /say, so be careful with your phrasing.)

```
/like /say I really like $1 !!!
```

...WILL work just fine.

This is true of ALL strings. To make one particular case easier, specifying channels, a few special strings have been created that automatically prefix the parameter with the # character. These are # \$1, # \$ \$1, and # \$?. See below for details.

In SOME cases you CAN do: /like /say I really like \$1's topic, but keep in mind: NOT ALWAYS !! You have to test this in every case you want to use it...

Where can you use Aliases?

Aliases can be used in the edit box (the line or lines you normally use to type commands in), and you can even use aliases in the Pops and the Remote sections that will be explained in the following sections. The function key aliases can also be used on selected nicks in the channel names-list and in query windows, etc....depending on your alias definitions. It is even allowed for aliases to call other aliases, or even to call themselves recursively. As a safeguard against infinite loops, the default maximum recurse depth is set at 5. This can be changed with the /maxdepth command. The minimum depth allowed is 1 (aliases can't call other aliases) and the maximum is 100. Calling aliases from within aliases uses quite a bit of memory, so this might cause problems in certain situations where memory is low etc. In case one of your aliases or remotes got into a loop use the CTRL+Break key combination to stop the process.

Special Alias Definitions ; The Function keys.

A special alias definition is possible which can launch commands using the Function keys. Define some aliases as follows:

```
/f1 /say Hello to you all
```

```
/f2 /topic # This is a new topic
```

You now can type "/f1" to say hello to all on a channel ... but also pressing the "F1" function will work !! Other examples:

```
/f5 /me will happily send you the new mIRC.  
/f6 /join #mirc  
/f9 /dcc send $1 c:.zip  
/f10 /dcc send $1 c:s.exe  
/f11 /dcc send $1
```

The F9 key will send the mIRC FAQ to a person selected from the channel names list. Don't forget to select a nick first ! If you don't, you will see strange (but logical) things... You can even use the F9 key in a Private, Query or DCC Chat window with this alias definition ! The F11 key will prompt you for the nick to whom to send a file... Well.... invent your own custom use of the F keys have fun ! You can also use -shift- and -ctrl- in combination with the F keys. Just define aliases like :

```
/f1 /say I used F1 to say this  
/sf1 /say I used Shift+F1 to say this  
/cf1 /say I used CTRL+F1 to say this
```

Using evaluation brackets.

You can now use [and] evaluation brackets in aliases to control the order of evaluation of identifiers. The brackets can be used in popups and remotes as well. Identifiers within brackets will be evaluated first, from left to right. You can also nest brackets. This allows you to evaluate things any way you like... The brackets will not show in the resulting text or command. Example's are :

```
/note /notice $1 " $+ [ $read [ -s + [ $2 ] ] c:\text\note.txt ] $+ "  
/sayvar /say % [ $+ [ $1 ] ]
```

Definitions without evaluation brackets are evaluated the same way as before.

Aliases in scripts.

You can easily load additional alias files into mIRC on top of the one(s) you already have. Open the aliases editor (Tools/Aliases/) and goto File/Load/. It is easy to select a file with aliases now. Select 'View' to see all loaded alias files. This is an easy way to try the aliases your friends created and to give your file with aliases to your friends. Aliases can even be included in a complete script file and distributed together with Remote Commands and Events. We'll explain this in detail later on.

7-1-2 String Definitions.

By now, you should have a decent grasp of the Aliases section. You might be interested in a full list of the strings and identifiers that you can use in your aliases and, later on, in popups and remote definitions as well, by now? The next two sections will introduce the most important ones. At this point, make sure to try all of the examples given until you fully understand how to do it yourselves! Always explore the help file for more information and the newest additions and changes!

\$N The simplest string. It contains one word. For instance \$1 will contain the first word after the alias command. By the same token, \$2 would indicate the second word, \$3 the third, etc. Example:

```
/op /mode $1 +o $2
```

Type "/op #mirc Janet John" to make Janet and John operator on #mirc.

```
/slap /me slaps $1 with a large $2
```

Type `"/slap residue fish"` to give residue some fish smell ;o)

\$N-M This string will only contain a selection of words. Try the alias:

`/three /say The first three words were $1-3`

Type `"/three a nice start kiddo"` to see the resulting "The first three words were a nice start" where "kiddo" is ignored. Note: The \$N-M replaces the old *N-M definition.

\$N- This string can contain a LOT of words... It will contain the first word, *and* everything after the first word. Try:

`/tell /say I'd like to tell all of you something - $1-`

The command `"/tell mIRC is really great"` with this alias will result in the text: "I'd like to tell all of you something - mIRC is really great". \$2- would contain the second word and everything after it. \$3- would contain the third word and everything after it. etc. Also try...

`/note /notice $1 $2-`

...and type `"/note <nick> <whatever text>"` And...

`/mess /msg $1 $2-`

...and type `"/mess <nick> <whatever text>"`

(Make sure you realize that these commands would have given the same result if you defined the aliases as `'/note /notice $1-'` and `'/mess /msg $1-'` respectively.)

Note : you can also use constructions like \$1-4

\$\$1 An alias containing this kind of string will be executed ONLY if a parameter is given; otherwise, the call to the alias will be ignored. For example, if you try this alias:

`/j /join $$1`

and then type, `"/j"` (instead of `"/j #channelname"`)... mIRC will ignore you because you didn't include a channel name. Otherwise, it works the same as \$1 -- it contains the first word. \$\$2 would contain the second word, \$\$3 the third, etc.

\$? This is a nice one!

An alias with this string in it can be started without specifying the parameter. mIRC will display a Parameter Entry dialog box, in which text, a nick, or whatever, can be entered. Try it with these examples:

`/j /join $?"`

`/hug /me hugs $? firmly`

\$?="text" An "extra special" version of the \$? string. mIRC will again display the Parameter Entry dialog box, but will now ask for the specific kind of information to enter. Try these:

`/j /join $?="Which channel to join?"`

`/hug /me hugs $?="Who to hug?" firmly`

\$! This identifier string is used only in conjunction with \$?. If you've used \$? earlier in the alias, you can use \$! Later on to recall whatever parameter was typed in. For example:

`/j /join $?="Which channel to join?" | /say I like channel $!`

Type `"/j"` and it will display a Parameter Entry dialog box asking you to type in the name of the channel to join. If you type `"/j #test"`, it will join #test, and then say into the channel: "I like channel #test". Also try:

`/friend /say $? is my friend | /say I like $! a lot`

##\$1 -and- #\$\$1 -and- # \$? These special versions of the \$1, \$\$1, and \$? strings are used in connection with channels only. All these do are affix the # character to the beginning of the parameter. Try...

```
/j /join # $1"
```

You'll be able to type "/j test" instead of "/join #test".

Also try:

```
/j /join # $?="Give channel name to join (No #)"
```

7-1-3 Identifiers.

A special type of string is the identifier. Where the content of a string is always defined by how an alias, popup or remote (see later) is called, the identifiers always will return pre-defined information. As with all other lists in this FAQ this list is not necessarily complete. Always explore the help file for more information and the newest additions and changes!

This most important identifier always contains the channel name where the alias was executed. This is important because many commands, like the /mode commands, require that you specify a channel name. Example:

```
/op /mode # +o $1
```

Type "/op Cindy", and it will op the person with the nick Cindy in whatever channel you type this.

\$me This pre-defined string always contains your current nick! You can use it in all kinds of situations.

For example:

```
/nice /me thinks $me is a nice nick !
```

```
/offer /me offers mIRC version 3.7 Do "/ctcp $me xdcc send #1" to get it.
```

```
/away /me is away ( $1- ) | /away $1- | /say Messages to $me will be stored.
```

Or give the alias:

```
/nick /say I like $1 better than $me | /nick $1
```

a try, and change your nick then by "/nick <newnick>"

\$read [-l#] <filename.txt> This predefined string is a funny one. It will pick a random line from the specified file and insert that line into the alias.... The line can contain plain text and, also, commands !! (to make it easier ;o) Also \$!read which is the same as \$read except that it isn't evaluated the first time it is passed through the alias routines. This allows you to use it in a /timer command if you want the \$read to be re-evaluated each time. For example:

```
/fun /say $read c:\mirc\funlines.txt
```

```
/kick /kick # $1 $read c:\mirc\funkicks.txt
```

```
/silly /say $read -l5 silly.txt
```

\$readini [-n] <inifile> <section> <item> Similar to \$read this identifier reads items from .ini files. It is created in conjunction to the /writeini command. Both use the built in windows routines to read from and write to standard Windows .ini files. We think \$readini and /writeini will open up new and better possibilities for writing your own automated actions on IRC. The -n parameter will prevent a line from being evaluated.

\$url This predefined string always contains the current page your www browser Netscape shows you... For example:

/show /say Hey guys, I found a superb www-page ! Check out \$url
/mu /me found this great URL : \$url

\$ip Your IP Address.
\$server The server you use.
\$active The name of the active window in mIRC.
\$time The time on your PC.
\$away Tells you if you're set away or not.
\$ticks Returns the number of ticks since your O/S was first started.
\$idle Returns same idle time as that returned by ctcp userinfo and finger.
\$online Returns the number of seconds elapsed on your online timer.
\$version Returns mIRC's version number
\$ctime The number of seconds elapsed since 00:00:00 GMT, January 1, 1970.
\$asctime(N) Returns ctime values in a full date text format.
\$day The current day. (Monday, Tuesday,...)
\$date The date (european order) (YES !! mIRC is -not- made in the USA ;o)
alternatively use \$adate (american), \$fulldate or \$day.
\$duration(seconds) translates numbers like 123456 into 1day 10hrs 17mins 36secs.
\$logdir, \$getdir, \$wavedir, \$mircdir, \$mididir,\$script and \$scriptdir return the paths to these directories.
\$mircini Returns path to and name of the ini file you're using.
\$nopath(filename) returns only the filename with no path.
\$nofile(filename) strips the filename and returns the remaining path.
\$exists(filename) Checks if the given filename exists. (Returning \$true or \$false)
\$findfile(dir,filename,N) searches the specified directory tree for the Nth specified file.
\$ddename The DDE service name mIRC uses.

\$abs(N) returns the absolute value of number N.
\$chr(N) returns the character with ascii number N.
\$asc(C) returns the ascii number of the character C.
\$len(text) returns the length of "text".
\$upper(text) returns "text" in uppercase.
\$lower(text) returns "text" in lowercase.
\$left(text,N) returns the left N characters of 'text'.
\$right(text,N) returns the right N characters of 'text'.
\$mid(text,S,N) returns N characters out of 'text' starting at S.
\$pos(string,substring) returns the position of substring in string.
\$replace(string,substring,replacement) replaces a substring in a string.
\$remove(string,substring) removes a substring from a string.
\$strip(text) strips out all bold, underline, reverse, and colour control codes from text.
\$count(string1,string2) returns the number of times string2 occurs in string1.
\$str(text,N) returns "text" repeated N times.
\$rand(X,Y) returns a random char inbetween X and Y.
\$lines(file) returns the number of lines in the specified file.

\$usermode returns your current personal mode.
\$nick(#,N) returns Nth nickname on channel #.
\$snick(#,N) returns Nth selected nickname on channel #.
\$opnick(#,N) returns Nth Op nickname on channel #.
\$nopnick(#,N) returns Nth non-Op nickname on channel #.
\$vnick(#,N) returns the Nth voiced nick on a channel.
\$nvnick(#,N) returns the Nth non-op/non-voice nick on a channel.
\$comchan(Nick,N) returns channels which both you and Nick are on.
\$query(N|Nick) returns the Nth query window you have open.
\$chat(N) returns the Nth dcc chat window you have open.

`$fserv(N)` returns the nick of the Nth file server window you have open.
`$send(N)` and `$get(N)` return the nick of the Nth open send/get window.
`$notify(N)` returns the Nth notify nick currently on IRC.
`$token(N,C,text)` returns the Nth token in 'text' separated by character C
(C is the ascii number of a character)
`$addtok(text,token,C)` adds a token to the end of text but only if it's not already in text.
`$findtok(text,token,C)` returns Nth position of token in text.
`$gettok(text,N,C)` returns Nth token in text.
`$instok(text,token,N,C)` inserts token into Nth position in text even if it already exists in text.
`$remtok(text,token,C)` removes 1 matching token from text.
`$reptok(text,token,new,C)` replaces 1 matching token with new token in text.

For the above identifiers you can specify N as 0 (zero) to get the total number of nicks/channels/etc. for that identifier.

`$snotify` returns the nick currently selected in the notify window.
`$address` returns the full address of the user that triggered a remote line.
`$address(nickname,type)` which scans the internal address list and returns a users address if it's found.
`$maddress` returns the currently matched user address for an event in the remotes.
`$maddress(address)` returns a matching address from the remote user list.
`$level(address)` finds a matching address in the remote users list and returns its level.
`$ulevel` returns the remote user levels that matched for an event.
`$clevel` returns the remote command levels that matched for an event.
`$dlevel` returns the remote default user level.
`$mask(address,type)` returns the address with the specified mask type.

Identifiers and variables can be placed inside the brackets.
Example: `/echo $right(3,$left($len(goats),ToMooOrNotToMoo))` results in ' Moo'

Identifiers with property parameters.

Several identifiers can be used with special parameters attached. This eases up remembering their syntax and might help a lot in scriping. The syntax is `$identifier(N|#|nick).property`
Always explore the help file for more information and the newest additions and changes!

`$server(N|nick)` Gives access to the servers list under File/Setup/IRC_Servers.
`$server(N)` for the address, `.desc` for the description, `.port` for the port, `.group` for the group.
`$ial(mask,N)` Gives access to the Internal Address List.
`$ial(mask,N)` for the full address, `.nick` for the nick, `.user` for the user, `.host` for the host, `.addr` for the address.
`$url(N)` Gives access to the urls list.
`$url(N)` for the address, `.desc` for the description, `.group` for the prefix
`$chan(N|#).property` Returns several 'channel' settings of channels which you're currently on.
`$chan(N|#)` returns channel name, `.topic`, `.mode`, `.key`, `.limit`
`$chat(N|nick)` nickname, `.ip`, `.status` (active, waiting, inactive)
`$fserv(N|nick)` nickname, `.ip`, `.status`, `.cd` (current directory)
`$get(N|nick)` nickname, `.ip`, `.status`, `.file`, `.size`, `.rcvd`, `.cps`, `.pc`
`$send(N|nick)` nickname, `.ip`, `.status`, `.file`, `.size`, `.sent`, `.lra`, `.cps`, `.pc`
`$timer(N)` Returns the timers you've active

```
$timer(N) for the id, .com, .time, .reps, .delay, .type
$group(N|name) identifier.
$group(N) will give the group name
$group(N).status and $group(name) tell if that group is on or off
```

User-definable identifiers. (Section originally written by Li0nheart)

As you just learned mIRC has a large number of built-in identifiers that you can use in your alias and remote definitions. It is also possible to make your own identifiers, and they can be configured to do virtually whatever you want! The identifiers are programmed in the alias section, and look exactly the same as an alias, with the only exception that at the end there is a /return X. You can have the identifier return whatever you want. For example, the sum of two numbers:

```
/sum {
%temp = $1 + $2
return %temp
}
```

Now you can use the \$sum identifier anywhere in aliases, popups, or remote. For example in the popup item:

```
Add:/say The sum of 45 and 78 is $sum(45,78)
```

Or the Alias:

```
/add /say The sum of $1 and $2 is $sum($1,$2)
```

(btw. it makes no sense to give a command like /sum .. that wont work)

All parameters given to the identifier are seperated with commas. You can use all usull if/then/else structures to in the identifier definitions. You could for instance check if all the required parameters are there, and return an Error command instead. For example, this routine that calculates what percentage X is of Y.

```
/perc {
if ($1 == $null) || ($2 == $null) { return Error not enough parameters }
if ($2 != 1) { %half = $2 / 2 } | else { %half = 1 }
%dummy = $1 * 100
%perc = %dummy / $2
%remainder = %dummy % $2
if (%perc == 0) { goto return }
elseif (%remainder >= %half) { inc %perc }
:return
return %perc
}
```

The first line makes sure all the parameters are there. The rest is calculation of the percentage, where part of it is used to make sure the total adds up to 100%. The identifier would be called with, for example, \$perc(56,100) which would display 56%. Numbers aren't the only thing you can return with an identifier. You can also make it return text, or combinations of text and numbers, for example, the time in american format (7:52 pm)

```
/atime {
set %hr $token(1,58,$time)
set %min $token(2,58,$time)
if (%hr == 0) { set %hr 12 | set %sub am }
elseif (%hr < 12) { set %sub am }
elseif (%hr == 12) { set %sub pm }
else { %hr = %hr - 12 | set %sub pm }
return %hr $+ : $+ %min $+ %sub
```

}

With a little creativity you can make identifiers for anything you want. For example, backwards talking, or highlighting every other letter. Have fun!

[Table of contents](#)

7-2 The Popups section.

Popups are the menus that appear when you right-click with your mouse. If you haven't found this nice feature of mIRC already, give it a try NOW! Some popup menu's are predefined in the popups.ini from the mirc distribution file, but they are fully customizable under the Tools/Popups/ menu item.

The mIRC Editor will start with the Popups section opened. Please ignore the Remote, Users and Variables sections for now! The mIRC Editor is used as a compact and uniform interface to almost all mIRC's programmable sections. With the Popups section selected the 'File' menu will help you to load, unload and save popup files. The 'Edit' menu offers copy/paste and search routines. 'Popup' shows an index of the current opened popup menu file.

The 'View' menu lets you select what popups you want to edit. There are different popup menus for different kind of windows in mIRC; In a Channel window you'll have a popup menu different from the popup menu in Queries (private conversations). Use the View menu to edit another popup menu. There are programmable popup sections for the Status window, Channel windows, Query/Chat windows and Nickname list and a special item called Menubar for the programmable Menubar item. (All explained later)

If your copy of mIRC doesn't seem to be equipped with some prefab popup menu's (the Tools/Popups/ section is empty then), it is almost certain that you didn't make sure that the prefab ini files were correctly copied from the mIRC distribution file to the mirc directory, or the place where the popups.ini file can be found by mIRC is not correctly specified. You can easily fix this in the File/Load/ menu item in the Popups editor.

The popup commands work almost exactly like aliases. All of the strings mentioned in Section 7-1 are allowed in popups (and vital for the proper use of them). If you haven't read the Aliases section (7-1), please do so now! You need to know that information before moving on.

Also: Definitely read the Help file's explanation of popups, as well as take a look at the many popup examples included in mIRC under Tools/Popups. The more examples you try, examine, and mess around with, the better your grasp of mIRC will be.

Example /join

Let's say that you join a particular channel frequently, #test. Rather than have to type in "/join #test", wouldn't it be easier to just right-click and select a menu item to join you there automatically? The following popup would do that. Just enter it into Tools/Popups, making sure that the "Status Window" is selected at the top. (Not Query/Chat window, Channel Names list or Menu Bar!)

```
Join the Test Channel:/join #test
```

Now right-click over your Status window and select "Join the Test Channel". mIRC will do exactly that.

Example /away

Here's an example of "away" and "back" popups... you type in the reason you're away, and it's sent to all the channels you're on. (That's what the /ame command does, it sends a /me to every channel you're on.)

```
Set Away...:/ame is AWAY ( $?="Give a reason" ) | /away Please try later ( $! ) !!  
Set Back:/ame is BACK | /away
```

(Note again that on some systems you may need to scroll to the right to see the full text of some examples.) Note that the "Set Away" example uses multiple commands.

Heirarchal Menus

If you were to add a few popups like that, your screen would quickly get cluttered every time you right-click. So you can organize your popups into heirarchal menus.

Try entering all of the following popups:

```
Say Things  
.Hello:/say Hello folks  
.Goodbye:/say Goodbye folks  
.Hug Somebody:/me gives $? a big hug
```

Here we're specifying that the last three commands should show up under "Say Things" by: 1) including them after "Say Things", in the order we want them to appear, and 2) prefacing them all with "." a period. When you right-click, you'll see the "Say Things" menu item. When you select that, you'll see "Hello", "Goodbye", and "Hug Somebody" menu items. Select from those to perform one of the actions specified above. If you want to get really complicated, you can make your popups up to four levels deep. For example, try:

```
Comments About People  
.Compliments  
..Nice Today:/say Wow, $? -- you're being very nice today!  
..Helpful:/say Thanks for being so helpful, $?  
..Friend:/say I'm glad to have $? for a friend.  
.Insults  
..Jerk:/say Leave me alone, jerk!  
..Idiot:/me wonders why $? is such an idiot sometimes.  
..Enemy:/say I'm proud to have $? as my enemy!
```

And so on! Just order them properly, using the periods. (If the manner in which the above is organized doesn't "click" with you, definitely try it out! Enter the above "comment" section into Tools/Popups. It's a lot easier to grasp when you see it working.)

All popup menu items can be put under a Function key too! That's handy as a replacement for often-used popups.... (Make sure to use the correct syntax and put them (ie.the function key replacements) in the alias section. Read more in section 7.1)

In the Tools/Popups section you will also find the new item that allows you to program the Menu Bar. Give it a try and put in or add the following lines to it:

```
Away  
.Set Away...:/ame is AWAY ( $?="Give a reason" ) | /away Set away at $time ( $+ $! $  
+ ) | /timer22 5 600 /describe # is away ( $! )  
.Set Back:/ame is back,.. lets start the fun ! | /away | /timer22 off  
Slaps  
.Slap!:/describe # slaps $1 around a bit with a frozen trout  
.Baseballbat:/describe # hits $1 on the head with his baseballbat !
```

[Table of contents](#)

7-3 The Remote Section.

The Remote section can be found under Tools/Remote/... . Clearly the most advanced part of mIRC, you can do things with mIRC's Remote section that would normally require advanced scripting or special bot software...

Things like: offering files (an XDCC list); responding to specific channel or private message text; customizing CTCP responses; setting up various user levels for your friends; ...almost anything that happens on IRC can be automatically responded to, and your response can make use of almost any mIRC command or commands.

The Remote syntax can be a bit complicated at times, but it's a simple extension of the formats explained previously for Aliases and Popups. If you understand those well, including the Strings, then please proceed. If not, go back and read (or re-read) Sections 7-1 and 7-2. The information presented there is ABSOLUTELY REQUIRED LEARNING before attempting Remote.

It's also important to remember why Remote is called Remote. The Remote section is designed to respond to things done by users OTHER than yourself -- in effect, it reacts to things happening on IRC around you and allows others to "remotely control" your mIRC session. All according to the specific Remote features you implement. Many examples can't be tested by simply sending the commands or typing the text yourself. They have to be sent by some other client or things have to happen on IRC.

The Remote section works closely together with the Users section. In the Remote section you'll define Scripts for the people in the Users section. Each User in your Users list can be assigned one or more levels. These levels dictate which events a user will be able to access or what happens if a users does certain things. In the (Remote) Scripts section you can define how mIRC reacts to things happening on IRC, in CTCP and (Raw) Event lines. By default the remote users list, variables and scripts are saved in the remote.ini file.

[Table of contents](#)

7-3-1 Remote Users.

In mIRC select Tools/Remote/ to open the mIRC Editor. Then make sure to select the Users section. In this part you can assign various user levels. You can assign certain levels to your friends and certain levels to your "enemies" (you can create a "shit list"), and a host of levels in between.

Why is this important? Let's say that you've written a Remote Event designed to kick anyone who says the word "nonsense" in your channel (just replace this in your head with whatever foul word you'd like). We'll explain how to do this later (in Section 7-3-3), but for now it's just the idea that's important. Sounds reasonable, right? OK, but perhaps you'd ONLY like regular users to be kicked for saying "nonsense." You'd like some of your friends only to be warned. And you'd like to ignore it from still others, perhaps your closest friends. And if you don't care for someone much at all, perhaps you'd like to kick AND ban them for saying "nonsense"! To do these kinds of things (give different responses to different people), you need to setup a users list. For each user, you'll specify their nick or address and the user level you've decided to assign them. Here's an example of a valid user list:

```
1:nick1
```

```
1:nick2
2:nick3!account3@machine.subnet.net
2:*!account4@machine.subnet.com
3:*!*@machine.subnet.edu
4:*!*@*subnet.edu
5:*!account@*.subnet.edu
10:yournick!youraccount@yourmachine.net
```

The numbers you pick for the user levels are completely up to you. You can decide what each of the user level numbers can and cannot do, as you'll see in the next two sections. By default, mIRC assigns everyone not specified in the user list to level 1. (Depending on the setting in Tools/Remote/Options/ 'Default user level').

If you specified different levels to different users you of course also need commands and events that have different reactions or power to users of these levels... That will be explained later...

As you can see, either a nick or an address are permissible. Wildcards are also allowed. There are some cases where you might want to specify your users twice, by both nick and address. These cases (with OP, DEOP, SERVEROP,.... and NOTIFY events) is explained in Section 7-3-3.

You don't have to do all the editing by hand ... you could also use mIRC's edit box or command line commands /user, /guser and /ruser.

/user <level> <nick or address> 'Auser' (Add user) adds whatever nick or address you specify to the users list at the level you specify. It does not check for a valid address. The nick or address is added exactly as you stated it.

/guser <level> <nick> [type] 'Guser' (Get user) adds a nick to the users list specified by address. To accomplish this mIRC does a /whois on the nick you specify, and adds the returned address on the specified level to the users list. This means that the specified nick has to be on IRC to be able to use the /guser command to add him/her to the list. By specifying the 'type' mIRC can add a user with several types of wildcarded addresses. This is analogous to the 'type' parameter used in the /ban command. Type ranges from 0 to 4.

/ruser <nick or nick! or address [type]> 'Ruser' (Remove user) removes a nick or address from the Users list. You must know exactly how the nick or address is stated in the User list for this command to work. If somebody is listed like nick!blabla@whatever.net you can also do "/ruser nick!". Mind the "!" to remove the entire line.

In mIRC an improved /ruser command is made so that it can work like /ban and /guser. /ruser <nick> [type] if you do NOT specify a type, then it works as usual and removes the specified nick from the user list. If you DO specify a type, it looks up the users address and removes that address from the user list.

Note: Using different address types allows you to specify a person or group of people with more or less wildcards. I'll show you the different results

```
If no type is mentioned type 6 is default; Kreet!*vonck@Wit399402.student.utwente.nl
type 0:*!vonck@Wit399402.student.utwente.nl
type 1:*!*vonck@Wit399402.student.utwente.nl
type 2:*!*@Wit399402.student.utwente.nl
type 3:*!*vonck@*.student.utwente.nl
type 4:*!*@*.student.utwente.nl
```

type 5;Kreet!vonck@Wit399402.student.utwente.nl
type 6;Kreet!*vonck@Wit399402.student.utwente.nl
type 7;Kreet!*@Wit399402.student.utwente.nl
type 8;Kreet!*vonck@*.student.utwente.nl
type 9;Kreet!*@*.student.utwente.nl

A remark; A user level can now even be a word, for example: mircop:!*
*@mardam.demon.co.uk (You can use this with events like on mircop:JOIN:#mirc:/mode
\$chan +o \$nick)

mIRC keeps an internal database of the people on the channels you are on. This database is used to speed up the performance of commands like /ban, /guser and /ruser a whole lot. These commands normally will make mIRC do a /whois on the person you issue the command on. The internal list stores info like nick!user@address for all users that are on the same *channels* as you. Whenever a /guser, /ruser, /ban, /finger, /ignore or /dns command is issued, the internal list is searched first to find the nick's address. If no match is found, the normal /whois stuff is done. This will speed up your bans a lot...especially when you're lagged... The 'Internal Address list' is activated under 'Tools/Remote/Options'. Switch it ON!

Table of contents

7-3-2 Remote Scripts - CTCP Commands.

In mIRC select Tools/Remote/ to open the mIRC Editor. Here you can set up how your mIRC reacts to things happening around you on IRC. Two distinct groups of reactions are possible; reactions to CTCP Commands and reactions to Events. In this paragraph the CTCP Commands are discussed.

Introduction - Remote Commands refers to a very specific type of commands: CTCP commands. CTCP stands for Client To Client Protocol. The Remote CTCP Command lines you can define in mIRC do one thing and one thing only -- they allow you to make mIRC to respond to CTCP commands sent by other users the way -you- wish.

There are a lot of ctcp commands defined according to the IRC RFC's (standards). The usual ones are version, ping, time, userinfo, clientinfo and finger. Every IRC client will respond in a standard way if one of these commands is send to it. These ctcp commands are sent by or to you in the format:

/ctcp <yournick> <command with parameter(s) if any>

Consider the simple example of a ping. The other user would type:

/ctcp <yournick> ping

(You can also send yourself a ping command! Try it! Type "/ctcp <yournick> ping".

Under normal circumstances, mIRC would send a standard ping reply [ctcp hisnick pong]. But with Remote Commands, you can redefine the program's response to the ping. It can now do almost whatever you'd like it to do when it receives a ping. On top of that, it can also send a normal ping reply. Or not. It's up to you.

There is ONE exception to this ... you cant hide your mIRC version reply.... We like the advertisement, you know...

You can also create new CTCP commands, and responses for them. This is how features like file offering (XDCC LISTs and SENDs) are handled. But right now, let's get back to our ping example.

Example: Custom Ping Response

Go to Tools/Remote and look in the 'View' menu if a script file called commands.ini or ctcps.ini is already there. If so, select that file and add lines to it and/or edit it. If not, go to the File menu and select 'New' to create a new script. Make sure that under "Listening" the CTCP's item is selected. (When items under "Listening" are not checked, for instance Remote Commands aka CTCP's, are ignored.).

Now place the following example into an existing or new script file.
`ctcp 1:ping:/notice $nick ping? Hmmm! ...pang pang pang!!`

Now if someone pings you, your reply will be "ping? Hmmm ! ... pang pang pang !!" in a notice sent to the person who pinged you, on top of the normal reply being 'pong'. Test this by giving the command `"/ctcp {yournick} ping"` If it doesn't work make sure your Remote Commands is set to 'Listening' ! Or enable it by giving the command `"/remote on"` and/or `"/commands on"`.

If you want mIRC to -not- give the default ping reply on top of your custom one you can block the default processing by the `/halt` command. Use it like

```
ctcp 1:ping:/notice $nick ping? Hmmm! ...pang pang pang!! | /halt
```

This will do the same thing as before, except it will not send your normal ping reply. Again, test it and see the difference!

Assuming you actually tried the two given ping replies, and that you got them to work, let's go over the syntax used here a bit, and then we'll go back to the above example and why it does what it does.

Remote CTCP Command Syntax.

Remote Commands, except for some special cases, are made up of 3 portions. Each of the portions are separated by a colon. The general syntax looks like:

```
ctcp <command_level>:<ctcp_command>:<triggered_command(s) with parameter(s)>
```

Lets go over the separate portions one by one:

`ctcp` This tells mIRC that a ctcp (Remote Command) definition follows.

`<command_level>` This command will respond to users with a user level equal to the command level, and to users with higher levels, unless other commands are created for them.

`<ctcp_command>` The CTCP command can be named anything you want it to be. It can be a standard, already existing CTCP command, like Ping or Version, or it can be one of your own creation. (like OPME, DIE, LEAVE, XDCC LIST,.....)

`<triggered_command(s) with parameter(s)>` In this part you define the (to be) triggered command(s). Their format and the use of multiple commands is almost exactly as you define them in the aliases and popups. The strings in this section are filled in by mIRC automatically... You only have to make sure that the strings you use are defined at all... ie. make sense if used in the remote commands ... (For instance the `$chan` is NOT defined in the remote commands !!)

Let's take a look at our ping example and try to dissect it.

```
ctcp 1:ping:/notice $nick ping? Hmmm! ...pang pang pang!!
```

The ctcp prefix tells mIRC a remote command follows. In the same scripts also aliases, events and raw definitions can be used. This will be explained later. 1 is the command level. All users with a user level equal or higher than 1 can execute or use this command.

Ping is the CTCP command this Remote Command is going to react to.

The triggered command is a /notice. In this case a notice send to \$nick containing the entire line "ping? Hmmm! ... pang pang pang!!" \$nick is a pre-defined parameter that contains the nick of the person who gave you the CTCP command ... in this case he pinged you. (More about the pre-defined parameters soon.)

Summarising, the effect of the command "/ctcp <yournick> ping" is a "/notice <nick> ping? Hmmm! ...Pang pang pang!!" This is executed as soon a CTCP Ping is received.

You can also use multiple commands, utilizing the | character, as with Aliases and Popups. That way a single Remote Command could activate multiple command lines. The use of the | is very straight forward, and we'll cover it later. Just glue two remote commands together using the |.

Different reactions to people with different user levels

You can fine tune your reactions to remote commands by assigning different levels to them. mIRC will always respond to a user with the highest leveled reply that fits to the command he/she gave.. Lets analyse it with these example lines...

```
ctcp 1:ping:/notice $nick ping? Hmmm your levels are tooooo low... | /halt
ctcp 2:ping:/notice $nick ping? Hmmm! ...pang pang .... Quite dead !!
ctcp 3:ping:/notice $nick ping? ...pang !! Hmmm... missed...
ctcp 6:ping:/notice $nick ping? Hmmm! ...Well I like you .. you're allowed to ping me..
```

A level 1 user won't get a ping reply but a nice "ping? Hmmm your levels are tooooo low..."

A level 2 user (2:nick3!account3@machine.subnet.net) will not survive it when he does a /ctcp <yournick> ping ;o) He will see "ping? Hmmm! ...pang pang Quite dead !!"

A level 3 user (3:*!*@machine.subnet.edu) will get " ping? ...pang !! Hmmm... missed..."

To level 4 and 5 users no specific reaction is defined ... The highest to them available command is the level 3 one... Therefore these people (5:*!account@*.subnet.edu) will see the level 3 reaction too !!

Level 6 and higher users will all get the level 6 ping reply...

This leveling can be done for ALL remote commands (and events) Make sure you get the point here !!! Maybe re-read the above sections ??

Special identifiers for Remote/Commands/ lines.

\$nick, \$address, \$site, \$level etc. are all identifiers specially created to be used in remote lines. All the usual String Definitions (7-1-2) can be used too. You can use them wherever you want in the lines triggered by remote commands...

You can use them wherever you want in the lines triggered by remote commands...

\$nick	The nick of the person who sent the command or activated the event.
\$address	The full address of the person who sent the command.
\$site	The site of the person who sent the command.
\$wildsite	Returns the address of the user who triggered the event in the form *!
*@host.domain	

\$level Represents the users remote level.

Note: The \$chan is NOT ALWAYS DEFINED in the Remote Commands ... \$chan normally contains the channel from which a commands is given ... CTCP commands don't have to be given to you from a channel ... or the person sending the command could be on more channels ... Get the point ?? Maybe \$active will do what you want ? Or \$target ?

Example: Here's an example using the pre-defined \$address parameter...
ctcp 1:address:/notice \$nick Your address is \$address

If a user were to send a '/ctcp <yournick> address' command to you, you would reply with a notice reading, "Your address is {whatever his or her address is}".

Example: Opping friends on demand.

If you're not clear on the basics of how all of this works yet, don't worry. Try to follow along with the examples and see if it "clicks" in your brain. Re-read the above examples if necessary. But don't attempt to write your own Remote Commands or move on to the next section until it "clicks."

```
ctcp 1:opme:/notice $nick Sorry, you won't get opped on $2 .. your level is too low
ctcp 5:opme:/mode $2 +o $nick
```

If a level 5 friend sends '/ctcp <yournick> opme <#channelname>', this command will op them in the specified channel. Level 1 people WON'T get opped whatever they try.... Here the \$2 parameter is used which refers to the second word the user typed in the CTCP opme command. In this case, it's the channel name. (\$1 will be the word 'opme')

The \$2- can be used to process long lines given by a remote user in a CTCP command...

```
ctcp 5:return:/notice $nick $2-
```

Multiple Commands.

A Remote CTCP command can also trigger multiple commands....

```
ctcp 5:opme:/mode $2 +o $nick | /notice $nick You are opped on $2
```

Then your level 5 friend sends the command:

```
'/ctcp <yournick> opme #channelname' What happens??
```

In effect two commands are triggered by the Remote CTCP opme command...

1st "/mode #name +o friend"

2nd "/notice friend You are opped on #name"

Another CTCP command triggering multiple commands:

```
ctcp 10:part:/part $2 | /notice $nick I have left channel $2
```

And if a level 10 user sends you:

```
'/ctcp <yournick> part #channelname'
```

This command makes you leave a channel on request, and tells the requester you did so...

\$2 contains the channel name you are asked to leave.

\$nick is the nick of the person who asked.

Other Examples.

```
ctcp 1:ping:/notice $nick Please don't ping me.
```

...would respond with: `"/notice <nick> Please don't ping me."`
`ctcp 5:time:/notice $nick Time to get a watch.`

...would respond with: `"/notice <nick> Time to get a watch.`
`ctcp 10:whoareyou:/notice $nick I am who I say I am | /notice $nick And I say that I'm`
`YOU!`

...would respond with: `"/notice <nick> I am who I say I am"` followed by
`"/notice <nick> And I say that I'm YOU!"`

[Table of contents](#)

7-3-3 Remote Scripts - Events.

In mIRC select Tools/Remote/ to open the mIRC Editor, then select the 'View' menu item and select a script named "events.ini". When no events script is already available eventually load a prefab script ini file or create a new script file with File/New/. A new script is called script1.ini by default.

As you've seen, Remote CTCP Script lines only deal with CTCP commands. Remote Events deal with just about everything else that can happen on IRC. If that sounds daunting, don't worry. As explained earlier, these lessons are cumulative. If you understand Remote Commands, Remote Events will be a piece of cake. You've already learned the really difficult stuff.

Warning: In general mIRC can be sensitive to poor Remote Event syntax. Using incorrect syntax can cause a GPF (General Protection Fault), so be careful! If ever you suddenly experience stability problems with mIRC, this is the part to check first!

Remote Event Syntax.

The Remote Events are: TEXT, INPUT, JOIN, PART, KICK, OP, DEOP, BAN, UNBAN, INVITE, NICK, QUIT, TOPIC, SERVEROP, NOTIFY, UNNOTIFY, MODE, USERMODE, SERVERMODE and SMODE. Besides the TEXT event the similar ACTION, NOTICE, NOTICE, CHAT, SERV and WALLOPS are available for more specific text origins. New are the FILESENT, FILERCVD and NOSOUND events as well as CHATOPEN, CHATCLOSE, SERVOPEN and the SERVCLOSE event. (Did I miss one?) CONNECT, MIDIEND,....

Exactly what they do, and how to use them, will be explained shortly. First, some quick examples and an explanation of the syntax used.

The syntax is essentially the same as with Remote Commands. Here's a sample Remote Event reacting on TEXT:

```
on 1:TEXT:hello:#: /msg $nick Hello back!
```

Go to Tools/Remote and look if a script file called events.ini is already there. If not, go to the File menu and select 'New' to create a new script. Make sure that under "Listening" the events's item is selected. (When items under "Listening" are not checked, for instance the events you created are ignored by mIRC). Now place the above example into an existing or new script file.

This Remote Event is triggered when the word "hello" is found in any active channel. mIRC's response is to send a private message to the person who wrote it, saying "Hello back." Here's another example, this time using JOIN:

on 1:JOIN:#:/notice \$nick You are very welcome to \$chan

This one is triggered whenever a user joins a channel you're on. A notice is sent to them that says, "You are very welcome to <channelname>".

If you've made it this far, and understood Remote Commands, this should seem fairly simple to you.

Remote Events have either 3, 4 or 5 portions divided by colons. All of the events have 4 portions, except for TEXT which has 6, and QUIT and NICK have only 3.

All of the events *except* for the TEXT, QUIT and NICK events are formatted like so: on <event_level>:<event>:<where>:<triggered_command(s) with parameter(s)>

on This prefix tells mIRC an EVENT will follow

<event_level> This event will effect users with this user level, and any higher levels. Unless other event entries are created for them. Just like command_level in Remote Commands.

<event> The name of the event we're listening for. JOIN, OP, etc.

<where> #, #name. Where are we listening for this event? Either # (any channel) or the name of a specific channel (like #test). EXCEPTION: Not used with NICK or QUIT. Then just skip this part.

<triggered_command(s) with parameter(s)> The (to be) triggered command. Just like in Remote Commands except that we have a few different pre-defined parameters.

...and of course, the big exceptions here are TEXT, ACTION, NOTICE, SNOTICE, CHAT and SERV which work like so:

on <event_level>:<event>:<what_text>:<where>:<triggered_command(s) with parameter(s)>

on This prefix tells mIRC an EVENT will follow

<command_level> Same as above.

<event> Well, obviously, it's going to be TEXT or ACTION

<what_text> Specify the word or phrase to listen for. Like "hello" in the first example.

<where> #, #name, ?, *. Just like above, you can tell TEXT and ACTION to listen on all channels (#) or on a specific channel (like #test). But you can also tell it to listen to all of the private messages sent to you (?) or to both channel and private message text .

A TEXT or ACTION event can't listen for things said in DCC Chats or Server messages. The NOTICE listens for notices like the CHAT listens for text in DCC Chat connections and SERV for text in DCC File Server windows.

<triggered_command(s) with parameter(s)> Same as above.

The Events

Here are the various Remote Events that mIRC can respond to...

TEXT event Occurs when specified words/phrases are found in a channel or private message.

ACTION works just the same ...

Example:

on 1:TEXT:nonsense:#:/kick \$chan \$nick No nonsense!

This will look for the word "nonsense" in any channel and kick the user who said it.

Example:

on 1:TEXT:nonsense:#test:/kick \$chan \$nick | /notice \$nick Hello \$nick , you said nonsense' and that's not allowed on #test...

If the word "nonsense" is said in #test, the sayer is kicked (if you're 'op' on #test) and sent a notice: "Hello <nick> , you said 'nonsense' and that's not allowed on #test..."

Example:

on 5:TEXT:help:?:/notice \$nick I wish I could help you, but I can't. If it finds the word "help" in any of your private messages, it replies to the user, "I wish I could help you, but I can't" in a notice.

You can specify wildcards in the text mIRC should react to:

- * reacts to any text
- word if a user said only (and exactly) this word the event will trigger. Exact match required.
- word* if a user started a line with this word the event will trigger.
- *word if a user ended any line with this word the event will trigger.
- *word* if a user said this word or phrase anywhere the event will trigger.

Example:

on 5:TEXT:help:?:/notice \$nick I wish I could help you, but I can't.

This event will only react to a call for help if the msg only contained the word help ... not if a line containing the word 'help' was sent to you... Example:

on 5:TEXT:hello*:?:/notice \$nick Hello to you too !

This event will only react to private message lines sent to you starting with the word 'hello'.

ACTION event See the TEXT event.... it works exactly the same....

This event will react to actions sent to you or the channel(s) you are on.

NOTICE event See the TEXT event it works exactly the same....

This event will react to notices sent to you or the channel(s) you are on.

CHAT event Works like TEXT but listens to a DCC Chat text.

SERV event Occurs when specified words/phrases are found in the File server window.

These events can be used to listen to text said in DCC Chat and DCC File Server windows. They both work almost exact like the TEXT event. Almost (?) logically the place denotification {where} is missing.... An important note has to be made here You CAN NOT use different event levels for the CHAT and SERV event for different levels of users ... I know, this is confusing but can't be solved ... Due to the direct client to client handling of DCC's, once a connection is set up, mIRC doesn't look for the nick or address of the user ... sorry... just level 1 CHATs and SERVs. Example:

on 1:CHAT:blabla:/msg \$me \$nick said \$1- to you in a DCC Chat.

SERV event Works like ON CHAT but listens to DCC Fileserver text. Example:

on 1:SERV:get mirc:/echo server 6 To \$nick the mIRC zip is sent.

JOIN event Occurs when a user joins a channel. Example:

on 1:JOIN:#test:/notice \$nick Welcome to the #test channel!

Sends a notice saying "Welcome to the #test channel!" to anyone who joins #test.

Example:

on 25:JOIN:#cool:/kick \$chan \$nick You're not welcome here!

Kicks any level 25 user who joins #cool with the message, "You're not welcome here!"

PART event Occurs when a user leaves a channel. Example:
on 10:PART:#:/notice \$nick Hey, thanks for stopping by \$chan !

Sends a notice saying, "Hey, thanks for stopping by <channel> !" to any level 10 who leaves a channel you're on. Example:
on 90:PART:#:/msg \$chan Yahoo! \$nick is gone!

Sends a message to the channel saying, "Yahoo, <nick> is gone!" if any level 90 leaves a channel you're on.

CONNECT event This event which triggers when the end of an MOTD is reached after connecting to a server (same as the perform section). Example:
on 1:CONNECT:/echo Successfully connected to \$server !

DISCONNECT event Triggers when you loose the connection to the server.
on 1:DISCONNECT:/echo Ooops... you lost the connection.

KICK event Occurs when a user is kicked from a channel. Example:
on 100:KICK:#:/kick \$chan \$nick | /invite \$knick \$chan | /notice \$nick That person is my friend!

If any level 100 person is kicked from any channel, you kick the kicker off the channel and invite the kicked person back, and send "That person is my friend!" to the kicker. NOTE: \$knick is the parameter for the kicked nick in this event. Also note that this example shows multiple commands. It works just like in Remote Commands.

OP event Occurs when a user is opped.
Note: This event requires people to be mentioned by NICK in your Remote/Users list!
Example:
on 1:OP:#test:/notice \$opnick You were opped by \$nick . Welcome!

If someone is opped on #test, they're sent a notice: "You were opped by <opper>. Welcome!" NOTE: \$opnick is the parameter for the opped user's nick in this event.

DEOP event Occurs when a user is deopped.
Note: This event requires people to be mentioned by NICK in your Remote/Users list !
Example:
on 1:DEOP:#:/msg \$opnick Tough luck. You were deopped by \$nick

If someone is deopped, they're sent a message: "Tough luck. You were deopped by <deopper>. NOTE: \$opnick is the parameter for the deopped user's nick in this event.

Explaining the OP, DEOP, SERVEROP as well as the NOTIFY event an important remark should be made. Due to the fact that servers only report the nick's of the people who are opped, deopped or notified on a channel or IRC, the OP, DEOP, SERVEROP and NOTIFY events only work properly if the nicks of the people who are opped, de-opped or notified are mentioned in the Users list. Specifying even the full address (nick!account@machine.address.net) is NOT enough... mIRC won't distillate the nick from such a statement. The only exception to this is if you wish the event to affect everyone by giving it an access level of 1, or whatever you have set the default user level to. This is the only case where it is not necessary to have a user in your User list by nick.

If you want to give people a level fitting to your OP, DEOP, SERVEROP or NOTIFY events you have to put them in your Users list by nick, like:

3:friendasnick
5:friendbsnick
5:friendcsnick
10:friendbsnick!accountb@bsmachine.net

This construction can give you a good reason to specify people in your Users list by both their full address (wildcards are permitted) and their nick. This could then guarantee the proper reaction to an OP, DEOP or SERVEROP without giving ALL people around with by accident (?) the same nick the same rights as your friends (or enemies).

BAN event

UNBAN event With the BAN event you can easily auto-unban your friends. Example:
on 5:BAN:#tree:/mode \$chan -b \$banmask

The \$banmask will contain the entire banned address. Easy huh ? The \$bnick identifier refers to the nickname of the user being banned. This is not always filled since sometimes a nickname isn't specified with a banmask!

INVITE event

Occurs when you are invited to a channel. Example:
on 5:INVITE:#test:/join \$chan | /describe \$chan thanks you for inviting him!

If invited to #test, join the channel and use an action to say "<me> thanks you for inviting him!"

NICK event

Occurs when a user changes his or her nick. Example:
on 1:NICK:/notice \$newnick I thought \$nick was a nicer nickname!

Send a notice to anyone who changes nicks, "I though <your old nick> was a nicer nickname!"

NOTE: \$nick is the parameter for the old nick, and \$newnick for the (guess what?) new nick.

NOTE: The <where> portion is not used in NICK.

QUIT event

Occurs when a user quits IRC. Example:
on 2:QUIT:/notice \$me Time to party! \$nick just quit the IRC!

Whenever someone quits, send a notice to yourself: "Time to party! <Nick> just quit the IRC!"

NOTE: The <where> portion is not used in QUIT.

TOPIC event

Occurs when a channel's topic is changed. Example:
on 1:TOPIC:#:/msg \$chan Wow, I love the new topic!

Whenever a channel topic changes, sends a message to the channel saying, "Wow, I love the new topic!"

SERVEROP event

Occurs when a server gives operator status (back) to a user on a channel. Note: This event requires people to be mentioned by NICK in your Remote/Users list ! Example:

on 1:SERVEROP:#mirc:/mode \$chan -o \$nick | /notice \$opnick Sorry, serverops not allowed on \$chan

Whenever a person is opped on #mirc by a server, the opped person is deopped and receives a notice "Sorry, serverops not allowed on #mirc" Serverops normally occur after a netjoin when the two splitted nets exchange and update all current channel modes.

NOTIFY event

UNOTIFY event These events trigger as soon as people in your notify list join or leave IRC.

Note: This event requires people to be mentioned by NICK in your Remote/Users list !

You could use this event to do a /whois on the notified nick to make sure it is the correct guy... Remember that this event ONLY works on nicks mentioned in the remote/user list by NICK.... (like OP, DEOP and SERVEROP) This is due to the IRC protocols which don't pass through the address of the notified person... Examples:

on 1:NOTIFY:/notice \$nick If you are the real \$nick then please join #mine !

on 3:NOTIFY:/whois \$nick

on 5:NOTIFY:/beep 10 50 | /whois \$nick

MODE event

SERVERMODE event These events can force a certain set of channel modes.

!! The MODE event has changed format. It no longer enforces modes. \$1- is now filled with the mode change that was made. Use this event to react to channel mode changes. Syntax and examples :

on 1!:MODE:#name:/echo 6 Mode change by \$nick on \$chan !

(Beware of loops when more people on a channel enforce mis-matching modes!)

USERMODE event This event triggers when you change usermode.

VOICE event

These events react to people getting or loosing voice on channels.

DEVOICE event

on 1:VOICE:/notice \$nick Welcome in the speaking world!

on 1:DEVOICE:/notice \$vnick Hey, \$nick is allowed to speak here!

SNOTICE event

This event listens for server notices. Server notices tell you all kind of things happening at and with the IRC server you use. To receive them you have to set your user mode to +s (/mode #nickname +s). Most people don't need this setting... It is mostly ment for IRCOps and server maintainers. With this event you can filter the server notices send to you and you can filter them ie. prevent them from being displayed or automatically react to them. The event syntax is very much like the ON TEXT event...

on 1:SNOTICE:servers_text:/echo 6 status The server noticed you about \$1-

FILESENT event

This event will react to all successful DCC Send file transfers. You can use it to send you or the receiver a small message...

on 1:FILESENT:*.txt,*.ini:/echo Sent \$filename to \$nick (\$+ \$address \$+)

on 1:FILESENT:mirc4516.exe:/notice \$nick Have fun with the 16 bit mIRC

on 1:FILESENT:mirc4532.exe:/notice \$nick Have fun with the 32 bit mIRC

FILERCVD event

This event will react to all successful DCC Get file transfers. It will trigger as soon as a file is successfully retrieved to your harddisk. You can use it to send you or the sender a small message, or to start a helper application to view the received picture, play a sound file or read the text.

on 1:ON FILERCVD:*.txt,*.ini:/run notepad.exe \$filename

on 1:ON FILERCVD:*.wav:/wavplay \$filename

on 1:ON FILERCVD:*.gif:/run c:.exe \$filename

on 1:ON FILERCVD:*.jpg:/run c:.exe \$filename

on 1:ON FILERCVD:*.mid,*.voc:/run wplany.exe \$filename

on 1:ON FILERCVD:*.*/notice \$nick Thanks for the file!

SENDFAIL event

GETFAIL event These events trigger when a DCC transfer fails due to a lost connection or transfer time-out. Examples:

on 1:SENDFAIL:*.txt:/echo failed to send \$filename to \$nick
on 1:GETFAIL:*.txt:/echo failed to get \$filename from \$nick

CTCPREPLY event Listen for replies to ctcps. eg.:

on 1:CTCPREPLY:PING*/:/echo \$active Got ping reply from \$nick
on 1:CTCPREPLY:*mirc*/:/echo \$active Wow \$nick uses mIRC too!

MIDIEND event This event triggers when a midi finishes playing (but not if you stop it by playing another midi or using /splay stop). Example:

on 1:MIDIEND:/echo mIRC finished playing midi file!

INPUT event This event is triggered when you enter whatever text message into an editbox and press enter. You can process the line with a script or whatever you like then! /halt prevents the standard processing of the text. Example:

on 1:INPUT:/echo you entered \$1-

LOAD event Triggers when mIRC is started and scripts are loaded.

START event Triggers as soon as your (new) script is ready to go.

on 1:LOAD:/echo script succesfully loaded
on 1:START:/echo script succesfully started

With the LOAD event you can give commands when mIRC loads the scripts. To give initialization commands which are run whenever mIRC is first run (and loads scripts automatically) use the START event. The START section also runs after the LOAD section when a script is first loaded. If a script is loaded from within the remote dialog, auto-run commands are not executed until the dialog is closed.

Special identifiers for Remote/Events/ lines.

\$nick, \$address, \$site etc. are all identifiers specially created to be used in remote lines. You can use them wherever you want in the lines triggered by remote events...

\$nick	The nick of the person who sent the command or activated the event.
\$address	The full address of the person who sent the command.
\$site	The site of the person who sent the command.
\$wildsite	Returns the address of the user who triggered the event in the form *!
*@host.domain	
\$level	Represents the users remote level.
\$chan	The channel on which some event triggered a Remote/Events line.
\$target	The destination of the event (fi. in server notices)

Table of contents

7-3-4 Various flags you can use in front of Command or Event lines in Remote Scripts.

Finally some remarks have to be made to the various flags you can use in the Remote section. (With Commands and/or Events)

The ! flag. (Events only)

This flag makes that events are not triggered by things you do yourselves....

Example:

on 1!:JOIN:#mine:/notice \$nick Welcome here.

The = flag. (Events only)

This flag is used to prevent anything from happening.... you can use it to not bother higher level people with events meant for lower level people.

Example:

```
on 1:JOIN:#mine:/ctcp $nick VERSION
on 3:JOIN:#mine:=
```

Level 1 and 2 users will be versioned while level 3 and higher users will not see anything at all,... nor you. (In version 3.2 you had to set up an harmless event to do this.)

Example:

```
on 1:JOIN:#test1:/notice $nick Welcome to #test1
on 1:JOIN:#test2:/notice $nick Welcome to #test2
on 3:JOIN:=
```

Level 3 and higher users won't be noticed Welcome.... lower level users only on #test1 and #test2.

The + flag.

This flag makes a certain command available ONLY to users with the EXACT level of the command... Higher (and of course lower) level users can't use/access it !

Example:

```
on 1:JOIN:#mine:/ctcp $nick VERSION
on +3:JOIN:#mine:/notice $nick Welcome here.
```

All users EXCEPT level 3 users will be versioned on channel 'mine' ... Level 3 users will be welcomed...

Example:

```
on +5:JOIN:#mirc:/msg $nick Welcome level 5 user!
```

The ; and REM flag.

These flags can be used to switch off commands or events temporarily by 'commenting them out'.

Example:

```
;on 1:JOIN:#mine:/notice $nick Welcome here.
REM on 1:JOIN:#mine:/notice $nick Welcome here.
```

The * and @ flags.

These equivalent flags make mIRC to execute an event or command ONLY if you're operator on the channel the event or commands if used on.

```
on *1:JOIN:#mine:/notice $nick Welcome here.
on @10:JOIN:#mine:/mode $chan +o $nick
```

The me flag.

This flag is made to make sure you can make events only working if somebody with exactly the same address as you use does something. This other person normally would be your 2nd instance of mIRC running from the same PC. Keep in mind that with proper leveling of your users in the Remote/users section you never (?) need this flag.

Conditional flags.

In mIRC version 3.7 some additional flags for the EVENTS are introduced. With these flags you can make mIRC to take into account the level of the person that started an EVENT. These flags can only be used with the OP, DEOP and KICK events since these are the only events that deal with somebody doing something to somebody else.

The three flags you can add are $>$, $<$ and $=$. But you can make mathematical combinations of them leading to a set of 6 flags: $<$, $>$, $<=$, $>=$, $<>$ and $=$.

These flags will make the event only be triggered if FIRST the level of the event fits the Opped, Deopped or Kicked guy (m/f) and then SECOND the level of the activator meets the expression $\langle \text{level-activator} \rangle$ [flag (mathematical expression)] $\langle \text{level-of-event} \rangle$

You look like you need some examples

Assume you being the GUARD with a Friend at level 2 and an Activator who can have several different levels.... Also imagine these simple EVENTS :

```
on <2:DEOP:#test123:/msg $chan the < triggered
on >=2:DEOP:#test123:/msg $chan the >= triggered
```

With the activator at level 1.

```
*** Activator sets mode: -o Friend
<GUARD> the < triggered
```

Your Friend has level 2, the Activator has level 1, $1 < 2$ is valid (one is smaller than 2), the first level 2 DEOP event is triggered...

With the activator at level 2.

```
*** Activator sets mode: -o Friend
<GUARD> the >= triggered
```

Your Friend has level 2, the Activator has level 2, $2 < 2$ is invalid, $2 >= 2$ is valid (2 is larger than or equal to 2), the second level 2 DEOP event is triggered...

With the activator at level 3.

```
*** Activator sets mode: -o Friend
<GUARD> the >= triggered
```

Your Friend has level 2, the Activator has level 3, $3 < 2$ is invalid, $3 >= 2$ is valid (3 is larger than or equal to 2), the second level 2 DEOP event is triggered...

As you can see its pretty straight forward once you got the idea read this over till the coin drops... More examples :

```
on <2:DEOP:#test123:/msg $chan the < triggered
on >2:DEOP:#test123:/msg $chan the > triggered
```

With the activator at level 1.

```
*** Activator sets mode: -o Friend
<GUARD> the < triggered
```

With the activator at level 2.

```
*** Activator sets mode: -o Friend
```

With the activator at level 3.

```
*** Activator sets mode: -o Friend
<GUARD> the > triggered
```

on =2:DEOP:#test123:/msg \$chan the = triggered

With the activator at level 1.

*** Activator sets mode: -o Friend

With the activator at level 2.

*** Activator sets mode: -o Friend
<GUARD> the = triggered

With the activator at level 3.

*** Activator sets mode: -o Friend

Table of contents

7-3-5 Variables

Select the Tools/Remote/ menu, and goto the "Variables" section ... In this part of Remote you can setup your own variables. Variables can contain numbers, words or even lines of text. On variables containing numbers you can apply mathematical manipulations to increase, decrease, add and subtract variables with given values to or from each other. Variables always start with a % and can have names of any length. The variables are stored between sessions in a variable file in the mIRC directory.

Several commands to create and maintain variables are available: (s=show; to force a status report of the operation to you)

```
/set [-s] <%var> [value] To create a variable  
/unset [-s] <%var> [%var2] ... [%varN] To delete a variable  
/unsetall To delete all variables  
/inc [-s] <%var> [value] To increase a variable by a value (number or variable)  
/dec [-s] <%var> [value] To decrease a variable by a value (number or variable)
```

The /set /unset /inc /dec commands all default to not displaying information to you, unless they are called directly from the command line. To force them to display info, you can use the [-s] switch eg. /set -s %x 1. With the aid of these commands you can make all kinds of handy aliases and remote lines! Let me give a bunch of examples :

Remote commands (see later) :

```
ctcp 2:xdcc send #1:/dcc send $nick c:\temp\serve\mir40.zip | /inc %mir40 1  
ctcp 2:xdcc send #2:/dcc send $nick c:\temp\serve\mirfaq24.zip | /inc %faq 1  
ctcp 1:stats:/notice $nick Sending stats: mIRC= %mir40 and FAQ= %faq
```

Remote events (see later) :

```
on 2:TEXT:xdcc send #1*:/dcc send $nick c:\temp\serve\mir392.zip | /inc %mir392 1  
on 2:TEXT:xdcc send #2*:/dcc send $nick c:\temp\serve\mirfaq23.zip | /inc %faq 1
```

Alias :

```
/stats /echo 6 mIRC: %mir40 and FAQ: %faq
```

With these lines I made a small download counter... not perfect but working !

You can even add some alias like the next one to keep partial statistics :

```
/reset /set %date $day $date | /inc %mirtot %mir40 | /inc %faqtot %faq | /set %mir40 0  
| /set %faq 0  
/stats /echo 6 Stats mIRC: %mir40 ( %mirtot ) and FAQ: %faq ( %faqtot ) (Set on
```

%date)

When you increase or decrease a non-existent variable it will be created automatically and set to the value you expect it to be. ie :

```
/inc %test1 6 will set %test1 to 6 if it didn't exist before  
/dec %test2 7 will set %test2 to -7 if it didn't exist before
```

Note : You have to make sure you always include the % ! This is made to prevent a lot of small possible problems in mIRC's command parsing. This does not limit the ways the variables can be used in at all since you can always use constructions like :

```
ctcp 1:upme:/inc % $+ $site 1  
ctcp 2:xdcc send #1:/dcc send $nick c:\temp\serve\mir40.zip | /inc % $+ $nick 1  
ctcp 2:xdcc send #2:/dcc send $nick c:\temp\serve\mircfq24.zip | /inc % $+ $nick 1  
ctcp 2:xdcc stats:/notice $nick You downloaded %nick files already since %date
```

You can also use an = (equal sign) to assign values to variables. Examples:

```
%x = 5 + 1  
%x = 5 - %y  
%x = %x * 2  
%x = %z / $2  
%x = %x % 3
```

[Table of contents](#)

7-3-6 Remote Scripts - Raw processing.

In the Tools/Remote Scripts mIRC also offers you the possibility to process all server <> client (mIRC) processes directly and in any way you want/like. It works exactly like the Remote Events except mIRC listens to the **numeric events**. These server numerics are described in the IRC RFC1459 (<http://ds.internic.net/rfc/rfc1459.txt>). More mIRC specific info is available on <http://www.teleport.com/~jeepster/numeric.html>

You better use Raw Script lines -only- if you exactly know what you do and only if absolutely needed. Wrong use of Raw Script lines can mess-up mIRC totally. It can over-rule all routines hardcoded into mIRC. Example:

```
raw 322:*mirc*/:echo 2 $1-
```

This would print all lines which have the word "mirc" in them when you do a channels /list (this is a pretty intensive test).

To echo a concise 2 line summary of the /whois return in your active window, place the following lines in your Raw section (listening on):

```
raw 311:*:echo 5 $active *** $2-  
raw 319:*:echo 5 $active *** $2-
```

This will display the user and channels lines. The numerics for the other /whois replies are: 312, server; 313, ircop; 301, away; 317, idle; and 318, end. For a full /whois echo, simply duplicate the above lines using these numbers.

To find out the numerics the servers use you could use the new \$numeric identifier that refers to the number of the (raw) event that was triggered. In the IRC RFC1459, Section 6; Numeric replies, you can look up all numerics, given with their number, name and reply string.

[Table of contents](#)

7-4 Advanced use of commands - Multi-line commands.

In mIRC you can use commands and create aliases, popups and remote scripts with conditional statements, loops and other tricky things. To explain it all to you would take another complete FAQ I fear, but I'll try to get you on the road in a short section with some examples.

I think its best to start right off the bat

First mIRC now allows multiline alias, popup, and remote definitions. This allows you to write statements in a nice and structured way like:

```
<prefix definition> {  
/command1 ...  
/command2 ...  
/command3 ...  
}
```

Example:

```
/away /ame is AWAY ( $+ $?="Reason" $+ ) | /away Set away at $time { $+ $! $+ }  
/back /ame is back,.. what did I miss ? | /away
```

These aliases could also be defined as

```
/away {  
/ame is AWAY ( $+ $?="Reason" $+ )  
/away Set away at $time { $+ $! $+ }  
}  
/back {  
/ame is back,.. what did I miss ?  
/away  
}
```

I agree that doesnt really help -here-, but while you're writing and testing the new possibilities in mIRC's conditional commands you might like this structure? Btw, the { } brackets are needed around all commands that use the new possibilitie of conditional looping etc. Now we're at it, you could leave out all / command prefixes if you wanted to... They are no longer needed.

A /goto command is added which can be used in { } definitions.

Example:

```
/greet {  
/set %x 0  
:retry  
/inc %x  
/goto %x  
:2  
/echo line2  
/halt  
:1  
/echo line1  
/goto retry  
}
```

This alias will echo the lines "line1" and "line2" to your screen.

Test it by giving this totally equivalent command in some editbox:

```
/set %x 0 | :retry | inc %x | goto %x | :2 | echo line2 | halt | :1 | echo line1 | goto retry
```

It will show you exactly what happens. ;-) You can also use a variable as a goto name, eg.
:%jumppoint

If you "/set %jumppoint 5" then you can do "/goto 5" and mIRC will evaluate %jumppoint to 5 and jump to it. In the above example the jumppoints were fixed to '1' and '2'. Try these 3 commands and see what happens now.

```
/set %jump1 1
/set %jump2 2
/set %x 0 | :retry | inc %x | goto %x | :%jump2 | echo line2 | halt | :%jump1 | echo
line1 | goto retry
```

You can use the /return command to finish processing a command and to allow any default processing to continue. eg.

```
on 1:JOIN:#mIRC {
  /echo 3 #mirc [Joins $nick]
  /return
  /echo 3 #mirc I'm not printed!
}
```

This will result in messages like:

```
[Joins henk]
*** henk (monster@ppp.dial.tip.nl) has joined #mIRC
```

Take care...

Take care not to get lost in incomplete bracket definitions. When opening brackets '{' are not closed '}' mIRC might start to behave in strange ways!! Work with care! A special check button is added to the Alias, Popups and Remote editor dialog to check if the brackets count in the current script is correct. It is the button with the "{}" picture on it.

/if /elseif /else

Now I think you're ready for the real stuff ?? In mIRC a simple /if statement is available

```
/if v1 operator v2 { ... } | /elseif v1 operator v2 { ... } | /else { ... }
```

Example:

```
Make this alias and start it like "/test 4";
/test { set %i 0 | :start | inc %i | if %i > $1 halt | echo $active %i | goto start }
```

If/elseif/else can all be nested inside each other. You should use () and {} brackets to make sure that you're terms are evaluated correctly (in the correct order), though you don't *have* to use them. Using brackets also speeds up processing quite a bit since mIRC then knows exactly what it has to evaluate.

Available comparisons and operators:

```
== equal to
!= not equal to
< less than
> larger than
>= larger than or equal to
<= smaller than or equal to
// is a multiple of
\\ is not a multiple of
```

```
isin string v1 is in string v2
```

iswm wildcard string v1 matches string v2
 ison nickname v1 is on channel v2
 isop nickname v1 is an op on channel v2
 isvo to check if user v1 has voice on channel v2
 isnum number v1 is a number in the range v2 which is in the form n1-n2 (v2 optional)
 ischan if v1 is a channel which you are on.
 isauto if v1 is a user in your auto-op list for channel v2 (v2 optional)
 isignore if v1 is a user in your ignore list with the ignore switch v2 (v2 optional)
 isprotect if v1 is a user in your protect list for channel v2 (v2 optional)
 isnotify if v1 is a user in your notify list.

To negate the above you can prefix them with an ! exclamation mark.

```
/if $nick !ison #mIRC { /echo $nick is NOT on #mIRC }
```

Example:

```
/massinvite { echo 4 * Mass-inviting # | set %i $nick(#,0) | :next | if $nick(%,%) !=
$me invite $nick(%,%) $1 | dec %i | if %i > 1 goto next | echo 4 * Mass-invite #
done }
```

Use this example alias like "/massinvite #yourchannel". (btw massinvites are very impolite!)

Example:

```
/randnoprk { :begin | set %kicknick $nick(#,$r(1,$nick(#,0))) | if %kicknick isop #
goto begin | /echo 6 %kicknick }
```

If you're an op on some channel you might like this random-non-op-kick. It kicks a random person from your channel, but never an op. Mind you.... if only ops are available you've a problem ;-)

Example:

```
/line { %line = "" | if $asc($1) < $asc($2) { set %i $asc($1) | :add | %line = %line
$chr(%i) | inc %i | if %i <= $asc($2) { goto add } | if (%line == "") { halt } | else
{ echo # %line | halt } } else echo # sorry not valid }
```

Start this alias example by typing "/line d k" to see what it does. It will print a line like 'd e f g h i j k'. Nothing much and barely useful but it shows the strength of the if/elseif/else pretty well.

Example:

```
/printnum1 { if $len($1) = 1 { if $1 !isin 1234567890 { echo 6 $1 is not a number |
goto end } } | elseif $len($1) = 2 { if $mid(1,1,$1) !isin 1234567890 { echo 6
$mid(1,1,$1) is not a number | goto end } | elseif $mid(2,1,$1) !isin 1234567890
{ echo 6 $mid(2,1,$1) is not a number | goto end } } | elseif $len($1) > 2 { echo 6 $1
has too many chars | goto end } | { set %x 1 | :begin | echo 6 %x | if %x >= $1
{ goto end } | else { inc %x | goto begin } | :end } }
```

```
/printnum2 { if $1 !isnum { echo 6 $1 is not a number | goto end } | elseif $1 !isnum
0-99 { echo 6 $1 is a too large number | goto end } | { set %x 1 | :begin | echo 6 %x
| if %x >= $1 { goto end } | else { inc %x | goto begin } | :end } }
```

These equivalent aliases will both print a list of numbers upto the value you give. Try it with "/printnum1 14" or something similar The second alias shows how smart use of identifiers can reduce the length of the alias a lot.

Empty or invalid identifiers

Variables or identifiers that don't evaluate to a value now return the value \$null so they can be used in the if statement for checking etc.

```
Example: /listops { echo 4 * Listing Ops on # | set %i 1 | :next | set %nick $nick(,#,%i) | if %nick == $null { goto done } | if %nick isop # { echo 3 %nick is an Op on # } | inc %i | goto next | :done | echo 4 * End of Ops list }
```

This alias will list all ops on the channel you're on.

Or in a remote event:

```
ctcp 1:CTCPREPLY:PING* {
  if ($2 == $null) echo [ $+ $nick PING reply]
  else {
    %pt = $ctime - $2
    if (%pt < 0) set %pt 0
    echo [ $+ $nick PING reply] %pt seconds
  }
  halt
}
```

More popup examples:

```
GiveOps { %i = 0 | %nicks = "" | :nextnick | inc %i | if ($snick(,#,%i) == $null) { if ($len(%nicks) > 0) mode # +oooo %nicks | halt } | %nicks = %nicks $snick(,#,%i) | if (4 // %i) { mode # +oooo %nicks | %nicks = "" } | goto nextnick }
```

This popup menu item will op all selected people on a channel where you're operator.

```
Selkick:/kick # $token($r(1,$snick(,#,0)),44,$snicks)
```

This popup will kick one random person out of the people you selected on a channel.

```
RandNopkick:/kick # $nopnick(,#,$r(1,$nopnick(,#,0)))
```

This popup will kick a random non-operator from the channel you're on.

```
RandNopkick { :begin | /set %kicknick $nick(,#,$r(1,$nick(,#,0))) | if %kicknick isop # goto begin | /kick # %kicknick }
```

This popup will also kick a random non-operator from the channel you're on.

```
Randkick:/kick # $nick(,#,$r(1,$nick(,#,0)))
```

Arent kicks fun ??? Another one ... randomly kicking just anybody might be -you- !

```
RandOpkick:/kick # $opnick(,#,$r(1,$opnick(,#,0)))
```

Randomly kicking an op they at least can defend themselves !

Endless loops.

With all these conditional statements I foresee you ending up in some endless loop. A simple example is creating something like:

```
/loop { :start | echo 6 # test loop | goto start }
```

I case one of your aliases or remotes got into such a loop use the CTRL+Break key combination to stop the process. It will take you home safely.

Custom windows.

In mIRC 5.0 a new tool is added that allows you to create your own custom windows. This window creation/manipulation tool can for instance be used to create a window in which you

can keep track of what your script does. A custom window is created with the /window command. It can handle a lot of parameters to define the sort of window that mIRC creates, the state of the window and to control its content.

```
/window [-abcdelnorsx] @name [x y [w h]] [/command] [popup.txt] [font [size]]
```

Switches:

a = activate window b = update horizontal scrollbar width for listbox
c = close window d = open as desktop window
e = editbox l = listbox
n = minimize window o = if opened on desktop, place on top
r = restore window s = use a sorted list
x = maximize window
@name = window name (must prefix with a @)
x,y,w,h = left top width height
popup.txt = popup filename, loaded when needed
/command = default command
font/size = font name and size (defaults to status window font)

You can also use /window to manipulate some of the above settings for an existing custom window. You can use the following commands to manipulate lines:

```
/aline [-cN] @name text    add line to list  
/dline        @name N        delete Nth line  
/iline [-cN] @name N text   insert line at Nth line  
/rline [-cN] @name N text    replace Nth line  
/sline        @name N        selects Nth line
```

Where -cN allows you to specify the line colour.

You can use the \$window(N/@name) identifier to access the following custom window information:

```
$window(N).x        left  
$window(N).y        top  
$window(N).w        width  
$window(N).h        height  
$window(N).state    minimized/maximized/normal
```

To access the lines in a custom window you can use:

```
$line(@name,N)      returns Nth line  
$sline(@name,N)     returns Nth selected line (listbox only)  
$sline(@name,N).ln to return line number of selected item.
```

This new tool offers a lot of usefull and creative uses but needs some experimenting to have it behave exactly as you want it to. Have fun!

[Table of contents](#)

