Visual IRC '97 Help

Welcome to **Visual IRC '97** version **1.00**, the leading freeware IRC (Internet Relay Chat) client for Microsoft Windows 95 and Windows NT featuring text, voice and video conferencing (see the DCC commands section), an advanced object-oriented scripting language (see **VSCRIPT.TXT**), and every built-in feature imaginable.

This help file has been extensively updated since the one included with ViRC '96 0.92b. It now covers every ViRC '97 feature in depth, including things like changing **colours and fonts** which were omitted from the ViRC '96 help file. **If you're new to ViRC '97, please read though the index page (that's this page) at least, as this may save you a lot of time that you would have spent wondering how to do something, or what a certain setting actually does.**

If you have a problem with ViRC '97, your first resort should be to go to the What to do if you have problems section.

Writing simple scripts in ViRC '97 isn't as difficult as people think. Introductions to writing aliases and events are included in this file - just page-down to the **Scripting** section below. However, it's thoroughly recommended that you read **VSCRIPT.TXT** before writing a more complicated script, even if you don't understand all of it just yet.

Quick overview of new features in 1.00 (over 0.92)

- Literally hundreds of new features ...
- New novice setup mode to guide new IRC users in detail through the setup process, from entering a nickname to joining a channel and speaking to people
- New Office '97-like user interface, including movable, resizable toolbars everywhere
- Hundreds of scripting enhancements
- Scripting and text output engines have been rewritten for a huge performance boost
- Built-in flood protection
- Support for language DLLs that support the Microsoft Active Scripting standard - this means that anyone with Microsoft Internet Explorer 3.0 or above can already use the VBScript and JavaScript languages to write their own scripts for ViRC '97 - see **AXSCRIPT.TXT**)
- Every bug reported has been fixed (hopefully :)
- New DCC Whiteboard feature for collaborative drawing and exchange of ideas
- Too many other new things to list!!

Also, please read the section on **command-line parameters** for information on enabling **debug**, **multi-user**, **exceptions**, **noaudio**, and **novideo** modes. And there currently isn't much documentation on **real-time audio** or the **DCC Whiteboard**, however, brief instructions on how to initiate a voice conversation and a Whiteboard connection are given in the DCC commands section.

Introduction and installation

Scripting

Shell integration

Epilogue

http://www.megalith.co.uk/virc and **http://www.virc.com**

What is IRC?

IRC is the Internet's chat protocol. IRC began in around 1989, and the protocol was designed by a group of guys from Finland.

To use IRC, you need an **IRC client**, such as **ViRC '97**, just as you need a WWW browser such as Netscape to access the web.

IRC is divided into a number of independent **networks**. The main two are **EFnet** (around 130 servers) and **Undernet** (around 30 servers). Both chat networks are totally separate (but you can be on both at once if you like!), however, all the servers in each network are linked together. It doesn't matter what EFnet server you're on - you'll be able to chat with people on any other IRC server transparently, without having to know what server they're on.

IRC offers two distinct approaches to talking to people. You may either talk to someone **individually**, or you may **join a channel** and conference with a group of people that way. All IRC channels begin with a **#** symbol (actually, channels present on only one server begin with **&**, but don't worry about that just yet).

IRC is sometimes used to talk about serious things, but most of the time, people IRC for fun. Also, don't be deceived into feeling forced to talk about things that relate to the channel name!! I'm a regular on **#quake**, and, although we talk about Quake occasionally, we actually talk about other things most of the time (although we're all Quake fans anyway).

IRC is doubtlessly a great way to meet people. There are two people I know who I especially admire, and I thought I'd give them a mention here. **BenJos** and **Denyse** actually met on IRC, fell in love, got married in January 1995, and have lived together happily ever since. In fact, BenJos emigrated from the Netherlands to the USA to live with Denyse. I can't promise you that kind of luck, of course, but I guarantee you'll have at least a little **fun** in the process!!

IRC clients are available for virtually every platform now. I'm not really in the business of promoting competitors' products, however, if you want to give other IRC clients a go, you could always go to Stroud's excellent **CWSApps** (which, incidentally, rates ViRC '97 above all other IRC clients) at **http://www.stroud.com/32irc.html**.

Installation

ViRC '97's installation is very simple and quick, and the product's system requirements are very low compared with many other products released today.

System requirements:

- **486** processor or above (**Pentium** recommended)
- Windows **95** or Windows **NT** (**3.51** or **4.0**)
- **8mb** RAM (more needed for NT)
- Just over **2mb** of hard disk space

Quick installation:

- Unzip the V97 distribution ZIP (called **virc96_*xxx*.zip**, where ***xxx*** is the version number) into any directory, using WinZIP or DOS pkunzip, which will extract the one file **INSTALL.EXE**.
- Run **INSTALL.EXE** to extract the ViRC '97 files to a directory of your choice.
- Run **VIRC97.EXE** to set up ViRC '97.

Setting ViRC '97 up if you've never used IRC before:

- Click on **I am new to IRC and require extended help** in the setup dialog, then click **Install**. ViRC '97 will take you step-by-step through everything you need to set up to get onto IRC, and will then guide you onto an IRC channel and explain some basic commands, describing everythnig in detail along the way.

Setting ViRC '97 up if you already use mIRC, WSIRC, PIRCH, or 16-bit ViRC:

- Click on **I use mIRC, WSIRC, PIRCH, or 16-bit ViRC** in the setup dialog, then click **Install**. You may be prompted to locate your IRC client's INI file. These are usually located in your **Windows** directory, but are occasionally found in the directory where you have the client installed.
- Your previous settings (personal details and server lists, but not scripts, which cannot be directly imported) will be imported automatically. ViRC '97 will then display the standard **User Setup** dialog to make sure your settings are correct.

Setting ViRC '97 up for the first time if you're familiar with IRC but don't use mIRC, WSIRC, PIRCH or 16-bit ViRC, or installing over a previous V97 installation:

- Select **New install, or upgrade previous version** and click **Install**.

The next lot of steps only apply if you're setting V97 up for the first time. If you're installing over a previous version of V97, all of your applicable settings will be preserved.

- First, decide on a **nickname**. Nicknames may be up to **9 characters long**, and may not contain spaces (use only **a-z**, **A-Z**, **0-9**, and the symbols **-** and **_**).
- Next, you must choose a **backup nickname**. Because there are so many users on IRC, do not be surprised if your chosen nickname is already in use!! The guidelines are to use the **same nickname here** as your primary nickname, but **adding** a _ (underscore) character to the end, or by **replacing** the final character with a _ if adding a _ would make the nickname

longer than **9 characters**.
- Third, enter your **email address** in the right field.
- Finally, enter your **real name** or **home page URL**.

After this procedure, click on **OK** in the **User Setup** dialog and ViRC '97 will be ready for use.

Why use ViRC '97?

Why use ViRC '97 over the **competition**?

Some IRC clients present the user with a **visually-appealing interface** which is **easy to use**. However, the actual client itself is not very powerful, and the experienced IRC user quickly becomes **frustrated** at the lack of scripting, or the limited customizability.

Other clients go to the opposite extreme. They include a powerful scripting language and are highly flexible, however, they appear very daunting, even hostile, to the average IRC user, and their built-in capabilities may be limited, as functionality is controlled by user-written scripts, which the novice would have no experience with.

ViRC '97 brings you the **best of both worlds** - and real-time audio chatting and video conferencing too. It features undoubtedly one of the easiest interfaces ever seen on an IRC application, and makes simple customization easy for even the most novice of users. However, beneath the surface lies the most powerful scripting engine ever written for an IRC application, **ViRCScript**, which experienced users can employ to extend the functionality of the client in any way they wish, releasing their scripts to other, possibly less experienced users for them to benefit too. ViRC '97 also includes a great deal of **built-in functionality** that is either not present in other clients, or only available through the use of add-in scripts. For example, ViRC '97 allows you to connect to **multiple servers simultaneously**, and also features a built-in **XDCC server** for creating an IRC file server, a feature only available in most other clients as an add-on script.

Notes on this help file

A number of **conventions** are used in this help file. They are as follows:

- Normal text - standard commentary on ViRC '97
- **Bold text** - for emphasizing words, or describing what you have to type or select
- *Italic text* - used for command parameters, and emphasization

Before we begin, it's important to note that most of the features described in this help file, including the menus, IRC commands, and text output, are **user-customizable**. It's possible that you may be using a script which makes everything look and work **totally different** from the manner described here!! This is the price you must pay for the extreme customizability ViRC '97 offers.

Connecting to a server

To start a new server connection, click on the **server window icon** on the main window's toolbar. This makes a new server window appear on your screen. The server that the window will connect to by default is shown in the **grey panel** in the top-right-hand corner of the window. To **connect** to this server, click on the **lit-up light bulb** icon on the server window toolbar. To **change servers** before connecting, double-click on the grey panel.

The connection status is displayed in the **right-hand pane** of the status bar at the bottom of the server window. You should see server connection proceeding in the following order.

1. **Resolving *servername* ...**
2. **Connecting to *servername* ...**
3. **Connected to *serverip:port.***

If the connection fails for some reason, instead of **Connected** appearing, you may see **Connection refused** or **Connection timed out**. If this is the case, the server may be down. Try another server!!

Note that you may connect to **multiple servers simultaneously**. Simply bring up a **number** of server windows, and connect to a **different server in each**. This is very useful if you want to be on, for example, EFnet and Undernet at the same time. This is faster and more convenient than starting up two copies of ViRC '97!!

Server window

Each server window has a **toolbar**. You can change what toolbars are displayed by clicking on the **system menu** (that's the server icon in the top-left hand corner of the window) and changing the settings from this menu. The toolbar buttons, from left to right, are as follows:

- Connect to server
- Disconnect from server
- Show channel box
- Show notify list pane
- Show server links
- Show channel list

The toolbar will be **easily customizable** in future versions of ViRC '97, although it's already possible to write a script to change the toolbar. See **OBJECTVS.TXT** for more information on this.

In addition, to rapidly change your server, just double-click on the **grey server panel** in the top-right-hand corner of the server window.

If you **right-click** on the server text area, a popup will appear. The default popup contains items such as cut & paste, and also options to connect, disconnect and signoff from the server.

If you **right-click** on the entry box, a popup will appear, which, as well as cut & paste functions, contains the option to change to a **multi-line edit box**. Some people prefer this format.

If enabled (by clicking on the **show notify list pane** button, which looks like a man with a tick), the **notify list pane** will appear to the right of the text output area. This will contain all the users in your notify list who are currently on IRC. You can right-click on any nickname in this list to perform a number of preset functions, such as doing a WHOIS on that user to obtain information about them, opening a private message (query) window with them, or sending them a file.

The server window also has a **status bar**. From left to right, the status bar shows **your nickname and usermode** (when you're connected), **transmit and receive lights**, the **lagscope**, and **status messages**.

The **lagscope** measures the amount of lag between you and the server (i.e. how long it takes for a packet to travel to the server and back again). Lag is displayed in seconds, i.e. **Lag - 0.32** means you are lagged 0.32 seconds (320ms) from the server. In addition, a **graphical indicator** is present behind the text display. The maximum lag the graphical indicator can show is **4 seconds**. Above this, and you should change to a different server!!

Channel box

ViRC '97 has a **channel box**, similar to, but more powerful than, mIRC's Channel Folder.

Initially, it is recommended you try out a number of channels using the JOIN command. After you've established a number of favourite channels, use the **channel box** to store them for easy recall.

Bring up the channel box by clicking on the **coloured # cube** in the server window's toolbar. The channel box has three **groups**, which can each be accessed by clicking on the appropriate **tab** at the top of the dialog box. These groups are:

- **Favourite channels**
- **Recently-accessed channels**
- **Web hyperlinks**

ViRC '97 automatically stores the last 40 channels you have visited in the **recently-accessed channels** group. This allows you to easily return to channels you have recently visited. It's very handy!!

In addition, you can store your favourite channels that you visit regularly in the **favourite channels** group. To add a channel to this group, simply type its name in the blue **channel entry field** and press **Add**.

There's a very simple way you can transfer channels from the recently-accessed group to the favourites group. Simply go to the recently-accessed tab, click on the channel you want to transfer, click on the favourites tab, and then click **Add**.

You can join a channel by clicking **Join**, or, even simpler, by **double-clicking** on the channel icon in the window.

ViRC '97 will automatically store any URL that's spoken anywhere in the **Web hyperlinks** folder. If you miss a URL before, it'll be there. Just press the **Go** button to go to that URL in your web browser.

Channel window

Each channel window has a **toolbar**, similar to that of the server window. You can change what toolbars are displayed by clicking on the **system menu** (that's the channel icon in the top-left hand corner of the window) and changing the settings from this menu. The toolbar buttons, from left to right, are as follows:

- Leave channel
- Cycle channel (leave and rejoin)
- Channel control
- Bold, underline, italic, and colour buttons

The toolbar will be **customizable** in future versions of ViRC '97.

Moving the mouse cursor over a **URL** in a channel window will make the mouse cursor change into a hand. Clicking the mouse over the URL will hyperlink to that URL with the currently-selected web browser defined in Hyperlinks.

If you **right-click** on the channel text area, a popup will appear. The default popup contains items such as cut & paste, and also options to change the channel mode and topic, if you are an op.

If you **right-click** on the channel names pane, a popup will appear. The default popup contains a standard, useful lot of nick functions, for example, WHOIS, query, various DCC things, and op functions, like kick and ban.

If you **right-click** on the entry box, a popup will appear, which, as well as cut & paste functions, contains the option to change to a **multi-line edit box**. Some people prefer this format.

If you **drag-and-drop** a nickname from the names pane to the entry box, ***nick*:** will appear in the entry box, ready for directing a channel message to that user. If you **drag-and-drop** a nickname from the names pane to the entry box while holding down the **shift** key, **/msg *nick*** will appear in the entry box, ready for writing a private message to that user.

Standard IRC commands

This section details the **standard set of IRC commands**, which appear in virtually every IRC client. More advanced commands (especially those pertaining only to ViRC '97) are detailed in a further section.

Notes: parameters that are compulsory are given like this: *parameter*. Optional parameters are given like this: *[parameter]*. Remember, don't type the *[ ]*'s! This is just notation!! Also note that, with some commands, an optional *[channel]* parameter can be supplied (see below). If *channel* is omitted, the command works on the channel window it's typed in. Otherwise, it is forced to work on the channel you specify, regardless of the window it's typed in.

JOIN command

Usage: /join *channel*

Joins *channel*, creating a new channel window. If *channel* doesn't already exist, it will be created for you, and you will be made operator. Examples:

```
/join #virc
/join #quake
```

PART command

Usage: /part *[channel]*

Leaves *channel*, closing the channel window. You must be on *channel* before leaving it!! Examples:

```
/part #virc
```

ME command

Usage: /me *[channel] text*

Produces an **action**. The following will appear in the channel window:

*** YourNick *text***

If /me is used in a query window, a **private action** is sent to the person you are querying.

DESC command

Usage: /desc *nick text*

Produces an **action** privately to *nick*. Same as /me, only sends the action **privately** rather than over a channel.

QUIT command

Usage: /quit *[reason]*

Signs off IRC cleanly, giving *reason* as your signoff reason. If *reason* is not specified, a default signoff reason (**Leaving**) will be used. Examples:

> /quit Back in 1 hour after The X-Files!!

TOPIC command

Usage: /topic *channel text*

Changes the topic on *channel* to *text*. The topic may be up to about 100 characters long. If the channel's mode is **+t**, you need to be a **channel operator** to use this command.

MSG command

Usage: /msg *nick text*

Sends *text* as a **private message** to *nick*. No-one else will be able to see the message except *nick*.

NOTICE command

Usage: /notice *nick*|*channel text*

Sends a **notice** to *nick* or *channel*. Notices are identical to private messages, only they may be handled differently by certain clients (e.g. displayed in a different manner). You would not normally have to use this command.

CTCP command

Usage: /ctcp *nick text*

Sends *text* to *nick* by **CTCP**. CTCP is used to query client-specific information. For example, to find out what IRC client **abc123** is using:

- **/ctcp abc123 version**

To query the current time where **abc123** lives:

- **/ctcp abc123 time**

QUERY command

Usage: /query *nick*

Starts a private message (**query**) session with *nick*. Any text you enter in the window is automatically sent to *nick* in the form of /msg's, and any private messages received from *nick* will be displayed in the window. This saves typing **/msg *nick*** every time you want to send *nick* a private message during a conversation.

AWAY command

Usage: /away *[reason]*

Sets you **away/here**. If *reason* is specified, you will be marked as away with *reason*, which will be displayed to any users who attempt to contact you while you are away. If *reason* is not given, you will be set as back.

SERVER command

Usage: /server *server[:port]*

Changes your IRC server to *server*. You may specify *port* if you want to connect to a port other than the default **6667**. You may specify the port either in the standard way (**server:port**) or in the mIRC-style way (**server port**).

WHOIS command

Usage: /whois *nick*

Queries the server for **user information** on *nick*. *nick*'s email address, channels, gecos (real name/URL), away status, IRCop status, and possibly idle time information are returned in a dedicated **WHOIS** window.

WHOWAS command

Usage: /whowas *nick*

Queries **user information** for *nick* if they have **just left IRC**. Most IRC servers only keep **WHOWAS** information for a few seconds after the user leaves.

UMODE command

Usage: /umode *[+|-][i][s][w]*

Changes your usermode. IRC servers support 3 usermodes which you can toggle, **i**, **s**, and **w**. The meanings of these usermodes are as follows:

- **+i** - you're invisible to **WHO/WHOIS** wildcard queries (use if you want privacy!!).
- **+s** - receive notifications of server-specific events, for example, when servers split or rejoin. You'll probably want to leave this off or your screen will fill up with useless junk.
- **+w** - receive messages directed at IRC operators.

IGNORE command

Usage: /ignore *nick|mask [[[+|-][all]]|[+|-]ctcp [+|-]public [+|-]msg]*

This complex command allows you to **selectively ignore different types of input from a user**. It is usually considered preferable to use the IGNORE command to deal with an annoying user, rather than using channel operator commands or IRC operator commands to dispose of them,

unless they are posing a threat to the channel or to the server.

Basically, IGNORE takes either the nickname of the user or their mask (for example, *! *megalith@*demon.co.uk) as the first parameter. Any addition parameters tell ViRC '97 what it should ignore or unignore. Examples:

- **/ignore greygoon** - ignores anything received from **greygoon**
- **/ignore greygoon all** - same as above
- **/ignore greygoon +all** - same as above

- **/ignore RayGamma msg** - ignores all private messages from **RayGamma**
- **/ignore *!*good_old_bill@*microsoft.com -public** - stops ignoring channel messages from that mask
- **/ignore *!*@* ctcp** - ignores CTCPs from everybody!! Aaaiee!!
- **/ignore MeGALiTH -** - unignores everything from **MeGALiTH**

Channel operator commands

You may use these commands only if you are an **op** on the channel.

MODE command

Usage: /mode *[channel] [+|-]mode [[+|-]mode ...]*

Changes the **mode** on *channel* to *mode*. The following modes are supported:

- **+o** *nick* - ops *nick*
- **+v** *nick* - gives *nick* voice (allow *nick* to speak on a **moderated** channel)
- **+b** *mask* - bans *mask* from joining the channel

- **+s** - makes *channel* secret (it will not show up in channel listings, nor will it show up in people's **WHOIS** listings).
- **+m** - makes *channel* moderated (only ops and users with **+v** may speak)
- **+n** - disallow channel messages from people not in *channel*
- **+i** - invite-only - people may only join *channel* on **invitation**.
- **+l** *number* - channel limit - only a maximum of *number* people may be present in the channel at any one time.
- **+k** *password* - sets the channel key to *password*. People can only join the channel if they know *password*.
- **+t** - only ops may set the channel topic.

Replacing **+** with **-** unsets the respective mode.

Don't use this command unless you know what you're doing! Misuse of **MODE** man lead to making the channel **unjoinable**.

KICK command

Usage: /kick *[channel] nick [reason]*

Kicks (forcibly removes) *nick* from *channel* with *reason*. You may use the *reason* parameter when kicking someone to give them an idea as to why they are being kicked.

BAN command

Usage: /ban *[channel] nick*

Bans *nick* from joining *channel*. A **ban mask** is generated in the form *!*user@*host for *nick* and is then set.

BK command

Usage: /bk *[channel] nick [reason]*

Kickbans *nick* from *channel* with *reason*. Equivalent to a **BAN** followed by a **KICK**.

FK command

Usage: /fk *[channel] mask [reason]*

Filterkicks *mask* from *channel* with *reason*. Anyone on the channel whose mask matches *mask* will be kicked. For example, to kick everyone off **#quake** who comes from the UK, you could use:

- **/fk #quake *!*@*uk**

FBK command

Usage: /fbk *[channel] mask [reason]*

Filterbankicks *mask* from *channel* with *reason*. *mask* is banned, and every user whose mask matches *mask* is then kicked. Equivalent to a **BAN** followed by an **FK**.

WALL command

Usage: /wall *[channel] text*

Sends *text* to all channel operators on *channel* in the form of a **NOTICE**. Note that you do not have to be a channel operator yourself to use this command (although walling from non-ops is always frowned upon), although you do need to be an op to see other people's walls.

IRC operator commands

ViRC '97 supports the full set of **IRC operator commands**. A few are detailed below, but not all of the them - IRCops should know their usage already.

OPER command

Usage: /oper *username password*

Makes you an IRC operator, using *username* and *password* supplied. If the server does not contain an operator record for you, you will receive a **No O-lines for your host** error from the server.

KILL command

Usage: /kill *nick reason*

Kills *nick* from IRC with *reason* (note that, unlike the KICK command, *reason* is compulsory here). IRCops who use KILL frequently are generally looked down on, so don't be KILL-happy!! If you have IRCop powers and dislike a user, use the IGNORE command instead unless the user is causing a disruption to the server, in which case a KILL is usually justified.

DCC commands

The **DCC** (direct client connection) protocol allows you to initiate a private, lag-free chat or file transfer with someone, without going through IRC (hence the word **direct**).

Current supported DCC types are **chat**, **send**, and **whiteboard**. Current supported TDCC types, specific to V97, are **send**, **voice**, and **video**.

DCC CHAT command

Usage: /dcc chat *nick*

Initiates a direct, secure **chat** session with *nick*. **0.63** and above feature a lag-indication feature in DCC Chats. Simply type **.** (a period) on a line of its own in the chat window, and, as soon as the remote end receives it, **PONG!!** will be sent back. That way, you can see whether your DCC Chat connections are lagged or not.

DCC SEND command

Usage: /dcc send *nick [file]*

Sends *file* to *nick*. If *file* is omitted, you will be prompted to select one or more files to send. If *file* already exists on the remote end's machine, you will be asked if you wish to **resume an interrupted transfer**. This will only work if the remote end's IRC client supports the resume protocol (currently, ViRC '96/'97, 16-bit ViRC, IaIRC, mIRC, and PIRCH support the resume protocol, but not many versions of ircII for UNIX). The DCC transfer window is now composed of a number of toolbars, which may be pulled out or rearranged to change the format of the window to whatever you want. You can turn toolbars on and off, and save the toolbar configuration, by clicking on the DCC transfer window's system menu (that's the icon in the top-left hand corner of the window). In addition, you can turn on and off the great DCC performance graph from here.

DCC WHITEBOARD command

Usage: /dcc whiteboard *nick*

Initiates a **DCC Chat with Whiteboard** connection with *nick*. When connected, both you and the other user can draw pictures and text on a single, shared drawing surface, and can also converse in a regular DCC Chat window. The DCC Whiteboard looks and functions like the Microsoft Paint application that's included with Windows 95 and NT, and you should be immediately familiar with what you can do. Currently, only ViRC '97, 16-bit ViRC, and IaIRC support whiteboard connections.

TDCC SEND command

Usage: /tdcc send *nick [file]*

Sends *file* to *nick*, using the **TDCC protocol**. TDCC file transfers may be up to **3 times faster** than DCC, however, *nick* must be using either ViRC, IaIRC, or a new version of PIRCH to receive files by TDCC. If *file* is omitted, you will be prompted to select one or more files to send.

If *file* already exists on the remote end's machine, you will be asked if you wish to **resume an interrupted transfer**. This will only work if the remote end's IRC client supports the TDCC resume protocol (currently, ViRC '97, 16-bit ViRC, and new versions of PIRCH support the TDCC resume protocol, but no versions of ircII for UNIX currently do).

TDCC VOICE command

Usage: /tdcc voice *nick*

Initiates a **real-time voice conversation** with *nick* using the **TDCC protocol**. Once the voice window has opened and connected, you may hold down the **Push to talk** button to speak to the other user. Release the button when you have finished talking. Alternatively, if you wish to speak for an extended period of time, you may click the **Lock** button to start talking, and you may click it again to finish talking. Currently, only ViRC '96/'97 support TDCC Voice connections.

If you have a **half-duplex** sound card and driver, you cannot speak and receive audio at the same time - audio reception will cut out when you are speaking. If, however, you have a **full-duplex** sound card and driver, both of you may speak and listen to the other user at the same time.

TDCC VIDEO command

Usage: /tdcc video *nick*

Initiates a **video conferencing session** with *nick* using the **TDCC protocol**. Once *nick* has accepted the connection, you will be able to video conference in a window. If you do not have a video capture card and camera, but *nick* does, your **Local** pane, and *nick*'s **Remote** pane, will remain blank, and vice versa. At the moment, video conferencing only works in **black-and-white**. If you, or the other user, have a colour camera, their video will appear in black-and-white only. This will be changed in a future release. Currently, only ViRC '96/'97 support TDCC Video connections.

More advanced commands

Listed here are some more advanced commands, many (but not all) of which are duplicates of functions already available on the toolbars and menus. However, they may be useful if you're writing a **script** that needs to make use of these functions.

MCI command

Usage: /mci *command*

Executes the **MCI** command *command*. MCI commands are used for playing or recording audio. For example, to play a sound file, you could use:

- **/mci play \windows\tada.wav**

To record a sound, you could use:

- **/mci record test1.wav**

And to stop recording (or playing) **test1.wav**:

- **/mci stop test1.wav**

In reality, you'll find that you only use **/mci play** to any great degree.

DNS command

Usage: /dns *hostname*

Performs a **DNS lookup** on *hostname* and displays the **IP address** it resolves to in the server notices window.

EXEC command

Usage: /exec *command*

Executes *command*. *command* can be a Windows or DOS program. You may also specify parameters to pass to the program if you wish. After the program has been started, it will be given the focus. V97 will **not** wait until program execution is complete before returning control - control will be returned immediately after starting the application.

USERADD command

Usage: /useradd *mask userlevel banlevel protlevel*

Adds *mask* to your **userlist** with the specified *userlevel*, *banlevel* and *protlevel*. ViRC '97's default events library attaches the following meaning to the levels:

- A **userlevel** of **1** or higher auto-ops the user when he/she joins a channel.
- A **banlevel** of **1** or higher auto-bankicks the user when he/she joins a channel.

- A **protlevel** of **1** or higher protects the user from being banned by someone else.

Your **scripts** may attach their own meaning to any higher levels. Specifying a *userlevel*, *banlevel* or *protlevel* of **-1** will keep the level at its current value. For example, to give everyone a *protlevel* of 1, but keeping the other levels the same, you could use the following command line:

- **/useradd *!*@* -1 -1 1**

CLEAR command

Usage: /clear

Clears the **output buffer** of the window it's entered in. You may find this useful if you wish to remove all the text from a **server** or **channel window**.

RLM, RLN and RLP commands

Usage: /rlm

Redirects the last private message you have received to the channel.

Usage: /rln

Redirects the last notice you have received to the channel.

Usage: /rlp

Redirects the last CTCP you have received to the channel.

WLM, WLN and WLP commands

Usage: /wlm

WALLs the last private message you have received to the channel.

Usage: /wln

WALLs the last notice you have received to the channel.

Usage: /wlp

WALLs the last CTCP you have received to the channel.

KLM, KLN and KLP commands

Usage: /klm

Kicks *user* with the last private message you have received as the kick reason.

Usage: /kln

Kicks *user* with the last notice you have received as the kick reason.

Usage: /wlp

Kicks *user* with the last CTCP you have received as the kick reason.

SOUND command

Usage: /sound *nick*|*channel soundfile*

This ridiculous command has been in mIRC for ages, and I've put a compatible version in V97 due to demand ... but I have no idea what use it is anyway. =] Anyway, /sound sends *text* along with *soundfile*'s name to *nick* or *channel*. *soundfile* must exist on both the local and the remote machines for this to work. If V97 receives a **SOUND** request, V97 will look for *soundfile* in its own directory only.

SPEAK command

Usage: /speak *text*

Speaks *text* through the Monologue speech synthesizer. Monologue is included with most Soundblaster cards and many compatibles. Monologue must be loaded for this to work (although this will change in a future version which will have built-in Monologue speech synthesis - more on this soon).

TITLE command

Usage: /title *text*

Adds *text* to the title bar. In fact, /title merely sets the value of the **$customtext** variable and then reprocesses the title bar's format string, as defined in the **Default** tab of **Client Setup**.

BACKGROUND command

Usage: /background <*none*>|*filename*

Changes the bitmap that's tiled over the main window's background to *filename* (which must be a **BMP** file). If <*none*> is specified as a parameter, the window's background image is removed (replaced with the standard grey background).

EXIT command

Usage: /exit

Immediately quits V97.

NEWSERVERWINDOW command

Usage: /newserverwindow

Creates, and sets focus to, a new server window. Equivalent to pressing the server window button on the main toolbar.

Client setup - Introduction

All aspects of V97 (except those that concern **scripting**) are configurable through the **client setup** dialog. Bring up the dialog by clicking on **Client setup ...** in the **File** menu. The different **setup pages** are accessible through the **tree structure** to the left of the dialog.

To get detailed help on a particular page, go to that page in ViRC '97 by selecting it from the tree structure on the left, select a control on that page, and then press the **F1** key.

Client setup - IRC settings

This tab of **Client setup** allows you to change your basic user settings.

Nickname

Enter here what nickname you wish to use on IRC. Nicknames may be up to **9 characters** long on most IRC networks, except for **DALnet**, where they may be up to **30 characters** long. Nicknames may **not** contain spaces.

Backup nick

If your primary **Nickname** is in use, V97 will automatically try what you enter here as an alternative nick.

Email

Enter your email address here.

Real name/URL

Some IRC clients call this the **gecos**. Here, you can enter any additional information you wish to make available to other users who query you with **WHOIS**. Most users enter their **real name** or the **URL** of their home page here.

Server

Allows you to change what server you wish to go onto IRC with. You cannot change this field directly. You must click on the small **combo down-arrow** button on the right to change the server, which will bring up the server editor dialog.

Usermodes

Here, you may set your default **usermodes** on startup. The meanings of the usermodes are as follows:

- **+i** - you're invisible to **WHO/WHOIS** wildcard queries (use if you want privacy!!).
- **+s** - receive notifications of server-specific events, for example, when servers split or rejoin. You'll probably want to leave this off or your screen will fill up with useless junk.
- **+w** - receive messages directed at IRC operators.

Client setup - DCC settings

In this tab of **client setup**, you may change settings that pertain to V97's **DCC** support.

Blocksize

Files are transferred block-by-block by V97. You may change here the size of each block that's sent. Generally, the larger the block size, the faster the transfer, however, blocksizes over about 1536 bytes (and sometimes a lot less) may cause file transfers to fail with **Operation would block** errors. If this is the case, reduce the blocksize until this no longer occurs. This setting applies to both **DCC** and **TDCC** transfers, however, the concept of blocksize works differently for **TDCC** transfers. You will find that increasing the blocksize will make very little difference to the file transfer speed, and you should not get **Operation would block** errors, no matter how large the blocksize.

Enable DCC send-ahead

Causes blocks to be queued and sent before the previous block has been acknowledged by the remote end. Although this works differently from the official DCC protocol, in practice, it works with every client tested and will yield much faster file transfers (although not **quite** as fast as **TDCC**). This is compatible with mIRC's **FastDCC**, although it works in a more efficient manner. Enabling this option should cause no problems.

Automatically accept DCC Chat requests

This option causes incoming **DCC Chat** connections to be automatically accepted without prompting the user whether he/she wishes to accept beforehand.

Automatically accept DCC/TDCC Send requests

Similar to the above, only automatically accepts **DCC Send** connections. Enable this option with **caution**, as files that already exist in your V97 **download directory** will be overwritten without confirmation if this option is enabled.

Automatically accept TDCC Voice requests

This option causes incoming **TDCC Voice** connections to be automatically accepted without prompting the user whether he/she wishes to accept beforehand.

Automatically accept TDCC Video requests

This option causes incoming **TDCC Video** connections to be automatically accepted without prompting the user whether he/she wishes to accept beforehand.

Automatically minimize DCC Chat window

Whenever a DCC Chat window appears on the screen, makes V97 discreetly minimize it to cause the minimum possible disruption.

Automatically minimize DCC/TDCC Send window

Whenever a DCC Send/Get window appears on the screen, makes V97 discreetly minimize it to cause the minimum possible disruption.

Automatically minimize TDCC Voice window

Whenever a TDCC Voice window appears on the screen, makes V97 discreetly minimize it to cause the minimum possible disruption.

Automatically minimize TDCC Video window

Whenever a TDCC Video window appears on the screen, makes V97 discreetly minimize it to cause the minimum possible disruption.

Close DCC transfer windows on completion ... if successful

Causes DCC/TDCC transfer windows to be automatically closed if the transfer has completed succesfully.

Close DCC transfer windows on completion ... if unsuccessful

Causes DCC/TDCC transfer windows to be automatically closed if the transfer terminates unsuccessfully.

Enable DCC forwarding to forward all incoming ...

Enables **DCC forwarding**. This feature allows incoming DCC/TDCC requests to be redirected to another nick. For example, if you wish to run ViRC '97 on your Windows PC, but receive files on your fast UNIX shell account, just load up ircII or a similar UNIX IRC client on your shell account and set the **DCC forwarding nickname** to the nickname you're using on the UNIX IRC client.

Forward all incoming ... DCC Chat requests

Forwards all incoming DCC Chat requests to the nickname specified.

Forward all incoming ... DCC/TDCC Send requests

Forwards all incoming DCC/TDCC Send requests to the nickname specified.

Forward all incoming ... TDCC Voice requests

Forwards all incoming TDCC Voice requests to the nickname specified.

Forward all incoming ... TDCC Video requests

Forwards all incoming TDCC Video requests to the nickname specified.

Client setup - Web browser (hyperlinks)

In this tab of **client setup**, you can configure how V97 interfaces with your **web browser**. V97 comes preconfigured with settings for **Netscape**, **Enhanced Mosaic** and **Internet Explorer**, so all you'll have to do is double-click on your browser to set it as the default.

If you select the **Automatic configuration** checkbox, you will not need to enter any additional information. Your web browser will be loaded up automatically, if it supports auto-loadup, which Internet Explorer and (probably) the latest versions of Netscape Navigator do.

Add

Adds a new browser to the list.

Remove

Deletes the currently-selected browser from the list.

Set default (or **double-clicking** on the browser name)

Sets the currently-selected browser as the default browser.

DDE topic

Sets the DDE appname and topic to be used when communicating with the browser. It takes the form *appname*|*topic*. *appname* is usually the browser's EXE filename, minus the EXE extension. *topic* depends on the browser, although browsers that support the Spyglass spec (which most do, nowadays) use **WWW_OpenURL**.

DDE item

Sets the DDE item to send to the browser when hyperlinking. This depends on the browser, although browsers that support the Spyglass spec would use something like **&,,0xFFFFFFFF,0x0**. V97 replaces the **&** in the item field with the URL to send before hyperlinking.

Client setup - IDENTD & Finger

This tab in **client setup** allows you to turn on and off V97's built-in **IDENTD** and **Finger** servers.

IDENTD on **UNIX** systems allows the server to use your real username for IRC, rather than what you supply to the client, which, of course, can be faked. However, IDENTD isn't supported intrinsically by Windows, so it must be supported by the IRC client. Turn this **on** unless you're already running an IDENTD server on your PC.

Most UNIX systems provide a built-in finger server, which allow other users to query information about you, and V97 offers the same sort of feature with its finger server.

Enable IDENTD server

Turns on V97's built-in IDENTD server.

Enable finger server

Turns on V97's built-in finger server. V97 will listen on port **79** for incoming finger requests.

Automatic configuration (IDENTD)

This will make V97 report via IDENTD the username you have entered in the user settings tab of **client setup** as the authenticated username. IDENTD will listen on port **113** for incoming connections, and report the system type as **WIN32**.

Username

If you have **automatic configuration** turned off, you may enter the username you wish to present to the IRC server here.

Port

If you have **automatic configuration** turned off, you may enter the port that V97's IDENTD server listens on. All servers connect to IDENTD on port **113**, so you should never actually have to change this.

System

If you have **automatic configuration** turned off, you may enter the system type ID that V97's IDENTD server presented to the IRC server. The RFC states that Windows 95/NT apps should present a system type of **WIN32**, however, many IRC clients (like mIRC) actually state the system type as **UNIX**. In reality, you should never have to change this.

Enter your finger text here

Allows you to change the **text** people will see when they attempt to finger you. Enter the text you want people to see in this edit box.

Client setup - Finger server

This tab in **client setup** allows you to configure V97's built-in **finger server**. Most UNIX systems provide a built-in finger server, which allow other users to query information about you, and V97 offers the same sort of feature.

Enable finger server

Turns on V97's built-in finger server. V97 will listen on port **79** for incoming finger requests.

Enter your finger text here

Allows you to change the **text** people will see when they attempt to finger you. Enter the text you want people to see in this edit box.

Client setup - SOCKS

This tab of **client setup** allows you to configure V97's **SOCKS support**. If you access the Internet through a firewall or proxy server, you can configure it here to allow you to access external IRC servers from within V97.

Enable SOCKS support

Tells ViRC '97 to connect to IRC servers through a SOCKS server. Note that DCC connections are not yet supported.

Automatic configuration

Enabling this option simply instructs V97 to automatically use the username you have entered in the user settings tab of **client setup** as your SOCKS username.

Username

If **automatic configuration** is disabled, you must enter the username you wish to use with the SOCKS server here.

Server

Enter the name or IP address of your SOCKS server here.

Port

You may enter the port number of your SOCKS server here if it differs from the default, which is **1080**.

Client setup - XDCC

This tab of **client setup** allows you to configure V97's **XDCC** support. XDCC provides an automated way for people to query and request files from you, much like a file server.

Once you have defined your packs, you can **offer your packs** to the channel by typing the **/xdcc** command.

Add pack

Adds a new **XDCC pack** to the list. You will be prompted for a brief description of the pack. Each XDCC pack may contain an unlimited number of files, and you may have an unlimited number of packs. However, it is recommended that you keep the number of files per pack down, as it is easier for other users to receive as few files as possible (if you have more than a few small files in the pack, it is recommended you ZIP the files and add the ZIP file to the pack, rather than adding the files directly).

Remove pack

Removes the currently-selected XDCC pack from the list.

Edit pack contents

Brings up the **XDCC pack editor** to allow you to modify the contents of the pack.
To **add files to the pack**, select one or more files from the file listbox on the left, and click on the **>** button. To add all files in the currently-selected directory to the pack, click on the **>>** button.
To **remove files from the pack**, select one or more files from the pack listbox on the right, and click on the **<** button. To remove all files from the pack, click on the **<<** button.
You may also change the **description** of the pack by changing the editbox at the bottom of the window.

Automatically offer packs to channels every ... seconds

When this checkbox is enabled, your packs will automatically be offered to all the channels that you're on at the regular interval specified.

Exclude these channels from auto-offer

Your XDCC packs will not automatically be offered to any channels entered here (separate multiple channels by a comma) when the **Automatically offer packs to channels every ... seconds** checkbox is enabled.

Send pack list to channels as NOTICEs rather than private messages

Normally, XDCC pack lists are offered to channels as regular messages. However, some people prefer them to be offered as NOTICEs, as this makes them appear different to other users and makes them easier to pick out.

Client setup - User lists

This tab of **client setup** allows you to configure your **userlist**. The userlist contains **user**, **ban** and **prot** entries, combining the functionality of friends, enemies and protection lists in other IRC clients.

V97 itself only acts on a userlevel/banlevel/protlevel of 1 so far, however, the userlist feature is script-extensible, so there may be scripts that you run which do different things to users configured with higher userlevels.

Add

Adds an entry to your userlist. You must enter entries in mask form (**nick!user@host**). The mask may contain ***** and **?** wildcards.

Remove

Removes an entry from your userlist.

Userlevel

This spin box allows you to set the user's userlevel. A userlevel of **1** or above will cause V97 to auto-op the user when he/she joins any channel that you're an op on. A green **U** will appear next to the entry.

Banlevel

This spin box allows you to set the user's banlevel. A banlevel of **1** or above will cause V97 to auto-bankick the user when he/she joins any channel that you're an op on. A red **B** will appear next to the entry.

Protlevel

This spin box allows you to set the user's protlevel. A protlevel of **1** or above will cause V97 to auto-unban the user whenever he/she is banned on any channel that you're an op on. A blue **P** will appear next to the entry.

Client setup - Ignore list

This tab in **client setup** allows you to configure V97's **ignore list**. The ignore list contains nicks and masks of people who you wish to ignore certain output from.

Add

Adds a new nick or mask to your ignore list. You will be prompted for the nick or mask to ignore.

Remove

Removes the currently-selected entry from your ignore list.

What to ignore ... Channel messages

Selecting this checkbox will ignore all channel messages from the currently-selected entry. A **#** will appear next to the entry.

What to ignore ... Private messages

Selecting this checkbox will ignore all private messages from the currently-selected entry. A **P** will appear next to the entry.

What to ignore ... Channel messages

Selecting this checkbox will ignore all CTCPs (including DCC requests) from the currently-selected entry. A **C** will appear next to the entry.

Client setup - Notify list

In this tab of **client setup**, you can specify a list of nicknames. V97 will periodically check these nicknames to see if they have joined or signed off IRC.

Add

Adds a nickname to the notify list.

Remove

Removes a nickname from the notify list.

Refresh

Forces the notify list to refresh. The status of each nickname is updated.

Auto-notify every 30 seconds

Selecting this checkbox causes V97 to check the notify list every 30 seconds, and fire the **<OnNotifyJoin>** and **<OnNotifyQuit>** events whenever any nick on the list joins or signs off IRC.

Client setup - Winsock

In this tab of **client setup**, you can see some information about the Winsock stack you are running, and change the buffer size.

Receive buffer size

The buffer size may be increased from the default 16k all the way up to 64k. If you are getting errors while doing a network-intensive operation (for example, a channel list or links list), increasing the buffer size may solve the problem.

Client setup - Window appearance settings

In this tab of **client setup**, you can adjust how V97 displays its windows.

Entry box height adjust

You can adjust the height of the entry box in V97 by a certain number of pixels by changing this value. If you use a small font for the entry box, you will want to set this to a negative value to reduce the height of the box, and vice-versa.

Border width adjust

This lets you change the width of V97's window borders. If you wish to **conserve screen space**, you can set this as low as **-2**, which removes all the borders, resulting in a "skinny" display, like mIRC. If you're running at a **high resolution** and like the chunky look of thicker borders and don't worry too much about screen space, you can increase this value as much as you want.

MDI window background

Select a bitmap file to **tile** over the main V97 window's background, in much the same way as you can specify a Windows wallpaper to be tiled using Control Panel. You cannot change the value of this field directly. You must press the **...** button and select a file. To remove the tiled bitmap from the background, press **space** or **delete** in this field.

No 3D border bevels

Turns the **window bevels** (3D borders) off, making the window edges appear **flat**. If you select this option, and you select **No 3D toolbar bevels**, and you set the border width to **-2**, the V97 windows will look just like mIRC's windows - flat and skinny. Some people prefer this look as it conserves screen space.

No 3D toolbar bevels

Turns the **toolbar bevels** (3D borders) off, making the toolbar edges appear **flat**. See **No 3D border bevels** above.

Client setup - Uninstall

Pressing the **Uninstall** button in this tab of **client setup** causes V97 to prepare itself for uninstallation. It will remove everything it has added to your registry, and exit. **Shell associations** will also be removed. It will not remove any files from your hard disk. In order to complete the uninstallation, you must drag the ViRC '97 directory into the recycle bin using Windows Explorer or delete the directory from an MS-DOS box using the **DELTREE** command.

Client setup - Directories

In this tab of **client setup**, you can change where V97 looks for different kinds of files.

Upload path

Sets the default directory where outgoing files are sent from. You cannot change this field directly. You must click on the **...** button to change the directory.

Download path

Sets the default directory where incoming files are received to. You cannot change this field directly. You must click on the **...** button to change the directory.

Script path

Sets the default directory that scripts are loaded from. When using the LOAD command without a path, the script will be loaded from this location.

SOUND path

Specifies where to find sound (WAV) files. This path is used as the default path for sending and receiving SOUND commands, and also for the MCI command to control multimedia functions.

Client setup - Server and channel options

In this tab of **client setup**, you can change a number of **options** that control the way ViRC '97 handles servers and channels.

Auto-connect to server on startup

Makes V97 open a new connection window and try to connect to the default server automatically on startup.

Auto-reconnect to server if connection failed

Makes V97 attempt to reconnect to the server if you managed to connect to the server but could not log on, for example, if the server is busy. If V97 cannot connect to the server, for example, if the connection is refused, V97 will not try to reconnect, regardless of this setting.

Retry to connect ... times

Sets the number of times V97 will try to connect before giving up.

Between connection attempts, wait ... seconds

ViRC '97 can be set to wait for a certain time interval to elapse before trying to reconnect to a server. This ensures that you don't flood the server with connection requests.

Show notify list pane in server windows by default

Setting this checkbox will automatically show the notify list pane in server windows. The notify list pane shows all the users in your notify list who are currently connected to IRC.

Open channel box automatically on connect

Brings up the channel box as soon as you connect successfully to the server?

Auto-rejoin channel when kicked

Automatically rejoins a channel if you're kicked off.

Seamlessly rejoin channel when kicked

If this option is set and you're kicked from a channel, do not close the channel window, but try to rejoin the channel seamlessly.

Show channel op control panel if you have channel ops

Shows the channel op control panel (containing Op, Deop, Voice, Devoice, Kick, and Kick and Ban buttons in a space-efficient format) if you have ops on that channel.

Close channel and qquery windows when disconnected from server

If this option is left unset, channel and query windows will not be closed when you disconnect from the server, so if you get unexpectedly disconnected, you can still read what has been said in your channel and query windows.

When connected to server, rejoin channels that you were previously on

With this option selected, when you connect to a server, the channels that you were on when you left that server will be rejoined. Great if you only ever use a small number of channels and want to automatically join them when you connect.

Client setup - ViRC '97 options

This tab of **client setup** contains a number of options relating to the way V97 appears visually.

Flash border(s) when minimized window contents change

Flashes the border blue and orange of some or all minimized windows when new text is added to them. You can control which windows should flash when new text is added to them by checking/unchecking the **Server windows**, **Channel windows**, and/or **Chat windows** checkboxes.

Enable button hover-highlighting

When selected, makes buttons light up when you move the mouse over them.

Enable URL hover-highlighting

When selected, makes URLs in server and channel windows become the selected text when you move the mouse cursor over them.

Display blank line at the end of text windows

With this option enabled, text output will appear like WSIRC, that is, there will be a blank line at the end of the text output box. Some people prefer this, however, it does take up extra screen space, so many like to turn the option off, which results in a mIRC-like text output display with no blank line at the end.

Totally hide or **Roll up drag-and-drop control centre on startup**

If **Roll up** is selected, the drag-and-drop control centre will appear rolled up to save space. If you'd rather do without it completely, select **Totally hide** and it will never be shown.

Hide main window status bar

Turns off the main window's status bar to save space.

Hide window tabs

Turns off the main window's tab bar to save space.

Toolbars initially rolled up to save space

Toolbars are initially hidden (press the yellow **magic dot** to make each one reappear) to save screen space.

Text buffer holds ... lines maximum

Sets the length of the text buffer. Increasing this value from the default of **100** will allow more text to be stored in each window, however, bear in mind that the larger you have this number the slower text output will be as it approaches this limit.

Enable graphical title bar

Turns on the graphical, gradiented title bar. Unless your display driver causes **glitches** when this option is enabled, if you like it, it should be left **on**. System performance is no longer degraded by this option.

Use dual-pane format for DCC Chat windows

Uses the old dual-pane (similar to Microsoft's Chat for IPX) format for DCC Chat windows rather than the standard single-pane format (as used by mIRC etc.).

Client setup - Miscellaneous options

In this tab of **client setup**, you can set miscellaneous settings which don't really fit into the other 2 groups (ViRC '97 options and Server and channel options).

Keep channel box and channel list open after joining a channel

Enabling this option prevents V97 from automatically closing the channel box and channel list windows after a channel has been selected and joined.

Use WHOIS dialog for command-line /whois queries

Selecting this option makes V97 use a WHOIS window to format and display output from command-line /whois queries. When the option is disabled, the WHOIS output appears unformatted in the server window.

Hide ping/pong display in server windows

Prevents displaying the ***** PING? PONG!** notice in server windows when you idle for too long. Selecting this option will not stop V97 from sending a pong back to the server, it'll just stop the display.

Use query window for private messages

Selecting this option will cause private messages from other users to go to individual windows, rather than displaying private messages in the server and/or channel windows.

Auto-minimize incoming query windows

Selecting this option makes V97 display incoming query windows minimized, so whatever you're doing isn't disrupted by incoming messages from other users.

Perform syntax check when loading ViRCScript scripts

Selecting this option makes V97 check the syntax of scripts while they're being loaded. This ensures that the script that's being loaded doesn't contain things like IF statements without corresponding ENDIFs, and so on.

Save window positions and sizes

With this option enabled, V97 will save the positions and sizes of windows when they're closed and restore them when they're opened again.

Accept DCC connections to ports below 1024 (leave off for security)

On UNIX systems, listeners on ports below **1024** can only be created by root. So all IRC clients are made never to listen for DCC connections on ports below **1024**. However, there are a number of system services below **1024** that are useful to people who wish to annoy you with war scripts. For example, port **19** just returns strings of ASCII characters as fast as it can - annoying if someone sends a **fake** DCC Chat connection from port **19** which you accept!! To

prevent this from happening, V97 will, by default, automatically **ignore** any DCC connections to ports below **1024**.

If you are writing something in ViRCScript which uses a DCC Chat connection to connect, for example, to a mail server, which listens on port **25**, you will have to turn this option off or your script will not work.

mIRC-style text window clipboard copy

Selecting this option makes cut-and-paste work like mIRC - just select an area of text in a window and release the mouse button to copy it to the clipboard. You don't need to right-click and select Copy when this option is enabled, as this will be done automatically after you've selected the text.

Automatically remove from received text ...

The options in this box choose what formatting codes ViRC '97 removes from incoming messages. **Bold attributes**, **Italic attributes**, **Underline attributes**, **Colour codes** and **Beeps** can be removed if desired.

Client setup - Automatic IRC responses

In this tab of **client setup**, you can set a number of default strings and options that V97 will use.

Automatically respond to ... CTCP FINGERs

Respond to **CTCP FINGER** requests.

Automatically respond to ... Version requests

Respond to **CTCP VERSION** requests.

CTCP FINGER

Sets the default reply text to send to users who **CTCP FINGER** you.

Version reply

Sets the default reply text to send to users who **CTCP VERSION** you.

Kick message

Sets the default kick reason if you don't specify one when doing the /kick.

Quit message

Sets the default signoff quote if you don't specify one when doing the /quit.

Title bar format

Sets the format of V97's title bar. It's safe to leave this alone (unless you want to change the format, of course).

Ban type

If you're a channel operator and you ban a user, this setting will determine how the user is banned. Usually, this can be left on the default setting.

Scripting - Aliases

To edit V97's **aliases**, select **Aliases...** from the **Scripting menu**.

This section does not detail **ViRCScript**, the language used to write aliases in V97. Rather, it describes how to implement simple aliases and goes through all the features of the **alias editor**. For a detailed reference on ViRCScript, see the file **VSCRIPT.TXT** that's included in the V97 ZIP.

What is an alias?

At the simplest possible level, an alias allows you to group a number of IRC commands together to be executed when you type one single command. Think of an alias as a single button you press to instruct the toaster to get to work, the kettle to boil and the fireworks to go off.

V97 displays a little ball beside each alias. The ball is normally green, however, if you have modifyed the alias that session, the ball will go red. This allows you to see at a glance which aliases you have tampered with if you're fiddling with aliases and something doesn't work any more.

Add

Adds a new alias. You will be prompted for the name of the alias you wish to add. If you want to make a new command **/go**, you could enter **go** or **/go** here - V97 ignores any leading **/** when adding aliases.

Remove

Deletes the currently-selected alias.

Name

In this field is the **name** of the alias (the IRC command you must type to activate it). You can change the name of the alias by editing this field.

Shortcut

Defines a **hotkey** which, when pressed, will trigger the alias, as if it were typed into the window that you press the key in. You define a hotkey simply by setting the focus to this field and pressing the key. Allowable hotkeys are **F1** to **F12**, and any other key with a **Ctrl** and/or **Shift** and/or **Alt** modifier. Note that you can't add normal alphanumeric keys as hotkeys. For example, you can't add a hotkey of **X** or **Shift+A**, for obvious reasons. Also, some keys might be reserved by Windows or by V97 itself. Don't add hotkeys of things like **Alt+Tab**, which is used by Windows to switch the focus to another application.

ViRCScript code for alias

Here you enter the ViRCScript code to execute when the alias is run. ViRCScript is a sophisticated, structured language, however, you don't need to know anything about it to start coding aliases. You can simply enter IRC commands straight in here. For example, to make

your alias **/go** join #abc123, set a topic there, and then signoff IRC, you could use the following code:

```
/join #abc123
/topic #abc123 Welcome to the #abc123 channel!!
/quit Be back later!!
```

As you can see, it's very simple to define aliases like this. Note that you must specify all channels explicitly. For example, if you left off the **#abc123** in the **/topic** statement line, it wouldn't work, as **/topic** would not know what channel you're referring to. Note that, in ViRCScript, it doesn't care about the preceding **/** on IRC commands. You can make the code more readable by dropping the **/** and capitalizing the first letter of the command. I would write the above script as follows:

```
Join #abc123
Topic #abc123 Welcome to the #abc123 channel!!
Quit Be back later!!
```

There are certain structured commands (see **VSCRIPT.TXT**), for example, **if** and **for**, that do not work with a preceding **/**. **All** commands work without it, however, so it's a safe bet to leave it off everywhere.

The variable **$C** is set to the channel you type the alias in. If you type the alias in a server window, **$C** is set to **.** (a period). For example, to **Say** some channel text into the channel you enter the alias in, you could use:

```
Say Hello everyone on $C!!
```

I've had people writing to me asking why their **Say** aliases don't work properly. When I looked at the code they were trying, they were using all sorts of **nonexistent** "variables" to represent the channel, for example **#**, **C#** and others. Remember, this is V97, not mIRC or PIRCH. >:->

See **VSCRIPT.TXT** under the **[2.2.1] Alias/EndAlias statements** section for more information on coding aliases.

Scripting - Events

To edit V97's **events**, select **Events...** from the **Scripting menu**.

Remember that event-editing is a feature for **advanced users** only!! Unless you want to add functionality to V97 with a script, and wish to respond to new events from the server or modify the handling of existing events, you shouldn't have to fiddle with this at all.

This section does not detail **ViRCScript**, the language used to write events in V97. Rather, it describes how to modify existing events and implement your own simple events and goes through all the features of the **event editor**. For a detailed reference on ViRCScript, see the file **VSCRIPT.TXT** that's included in the V97 ZIP.

What is an event?

Unlike every other IRC client, even ircII, V97 is totally **event-driven**. What this basically means is that V97 by itself has no idea how to interpret any data received from the IRC server. All text displayed by V97 is actually displayed by **events**, which are fired by V97 on reception of specific data received by the server. This provides a **very** powerful approach, as it allows custom scripts to be written which handle new events and implement new server features that V97 doesn't even know about yet.

Basically, an **event mask** is specified for every event defined. The event is fired if data receives from the server matches the mask. The mask is parsed **word-by-word**. Each word of the mask must match the corresponding word of the line received by the server. For example, the mask for the **JOIN** event is as follows:

> *** JOIN ***

The actual line received from the server when someone joins may be something like this:

> **:nick!user@host JOIN #channel**

In order for the event to be fired, the first word can be anything (the mask for that word is *****), the second word must be **JOIN**, and the third word can be anything. Any subsequent words which are not specified in the mask can be anything. For example, the mask for receiving a channel message is *** PRIVMSG #* ***. You do not need to specify *** PRIVMSG #* * * * * * *** etc... until all the words have been covered. V97 will handle this automatically.

Each event has a little symbol to the left of it. If the symbol says **ON**, the event is enabled. If it says **OFF**, the event is disabled and will never be fired. You can enable and disable events by clicking on the **Event enabled** checkbox.

Add

Adds a new event. You will be prompted for the **name** of the event (which can be any arbitrary text you like, although should be descriptive of what the event does, and may not contain spaces) you wish to add. You will then be prompted for the event's **mask**. If you want to make a new event for the server code **353**, for example, you could enter **ServerEvent353** for the name, and *** 353 *** for the mask.

Remove

Deletes the currently-selected event. Be careful when deleting predefined events!!

Name

In this field is the **name** of the event. You can change the name of the event by editing this field.

Mask

Enter the **mask** to fire the event on in this field.

Event enabled

This box is checked by default. Unchecking this box disables the event, that is, it will never be fired.

ViRCScript code for event

Here you enter the ViRCScript code to execute when the event is fired. ViRCScript is a sophisticated, structured language, however, you don't need to know much about it to start coding events. See **VSCRIPT.TXT** under the **[2.2.2] Event/EndEvent statements** section for more information on coding events.

Scripting - Menus/popups

To edit V97's **menus and popups**, select **Menus/popups...** from the **Scripting menu**.

This section does not detail **ViRCScript**, the language used to write code for menus/popups in V97. Rather, it describes how to modify existing menus/popups and implement your own simple popup enhancements and goes through all the features of the **menu editor**. For a detailed reference on ViRCScript, see the file **VSCRIPT.TXT** that's included in the V97 ZIP.

Introduction

Unlike every other IRC client, V97 has no built-in menus. All the menus are defined by a script, and a default, standard set of menus are defined by DEFAULT.LIB, which is loaded by V97 when you install it for the first time.

Menus and popups in V97 are defined in the form of **trees**, which is a logical way for menus to be designed. Imagine a typical application's **File** and **Edit** menus. A tree for the menus might look like this:

```
File
    |
    +-New
    +-Open ...
    +-Save as ...
          +--
    +-Exit

Edit
    |
    +-Cut
    +-Copy
    +-Paste
```

In V97, menus are defined in a similar way. When you define an item, you have to give it a depth, which his the depth of the item in the tree. In the examples above, **File** and **Edit** would have a depth of **0** (compare this to a plant's **root**) as they are base items, and all the subitems would have a depth of **1** (compare this to a plant's **branch**). You can, of course, also make subitems off items. For example, in your file menu you may have the option to make a new file in a different format:

```
File
    |
    +-New
          |    |
          |    +-Java source file ...
    |    +-C++ source file ...
    |    +-Plain text file ...
    +-Open ...
    +-Save as ...
          +--
```

```
        +-Exit
```

In this case, **File** has a depth of **0**, **New**, **Open ...**, **Save as ...**, **-** (the separator) and **Exit** have a depth of **1**, and **Java source file ...**, **C++ source file ...** and **Plain text file ...** have a depth of **2** (compare this to a plant's **leaf**). So the concepts behind menus are very, very simple as you can see (currently, V97 supports a depth of **0**, **1, 2, 3** and **4**. This should be enough for any practical purposes).

Tab bar

The tab bar selects what menu or popup you want to edit. You can edit the **Main menu**, the **Server popup**, the **Channel text popup**, and the **Channel nicks popup**:

- Main menu - V97's main window menu bar
- Server popup - right-click menu for server windows
- Channel text popup - right-click menu for channel text area of channel windows
- Channel nicks popup - right-click menu for nicks area of channel text windows
- Query popup - right-click menu for query windows

Add

Adds a new menu item. You will be prompted for the **name** of the menu item (which can be any arbitrary text you like, although it should be descriptive of what the menu item is, and may not contain spaces) you wish to add. You will then be prompted for the **text** as you want the menu item to appear. For example, if you wished to add an **Extras** menu to the main menu bar, you would click **Add**, and enter, for example, **M_EXTRAS** as the **name** and **Extras** as the **text**.

Remove

Deletes the currently-selected menu item.

Name

In this field is the **name** of the menu item. You can change the name of the menu item by editing this field.

Text

This field contains the **text** of the menu item, i.e. how the item actually appears on the screen.

Shortcut key

To assign a shortcut key to the currently-selected menu item, press the key in this field. Note you can only assign shortcut keys to non-top-level items in the **main menu** (e.g. a shortcut key cannot be assigned to the **File** menu itself, but a shortcut key can be assigned to any item in the **File** menu).

State

Here you can change the **state** of the currently-selectedmenu item. If the currently-selected menu is **Main menu** or **Server popup**, the **state** can take one of the following values:

0 - Menu item is enabled
1 - Menu item is enabled when connected to the server, otherwise enabled
2 - Menu item is disabled when connected to the server, otherwise enabled
3 - Menu item is disabled

If the currently-selected menu is **Channel text popup** or **Channel nicks popup**, the **state** can take one of the following values:

0 - Menu item is enabled
1 - Menu item is enabled when you're opped on the channel, otherwise disabled
2 - Menu item is disabled when you're opped on the channel, otherwise disabled
3 - Menu item is disabled

If the currently-selected menu is **Query text popup**, the **state** can take one of the following values:

0 - Menu item is enabled
3 - Menu item is disabled

Up arrow

Pressing the **up arrow icon** moves the currently-selected menu item up in the list. You can use this button, together with the **down arrow icon**, to easily move items about in a menu.

Down arrow

Pressing the **down arrow icon** moves the currently-selected menu item down in the list. You can use this button, together with the **up arrow icon**, to easily move items about in a menu.

Left arrow

Decreases the **depth** of the menu item. See above for a description of **depth**. Each level of depth is shown in the list by a **...** symbol before the menu item's **text** in the list.

Right arrow

Increases the **depth** of the menu item. See above for a description of **depth**. Each level of depth is shown in the list by a **...** symbol before the menu item's **text** in the list.

ViRCScript code for menu item

Here you enter the ViRCScript code to execute when the menu item is clicked. ViRCScript is a sophisticated, structured language, however, you don't need to know anything about it to start coding menu items. See **VSCRIPT.TXT** for more information on coding menu items.

Technical info about ViRC '97

Lots of people ask things like what language V97 is written in, and so forth, so I'll try to answer the most common ones here.

What language is V97 written in?

V97 is written totally in Borland Delphi 3.0 Professional.

How long has V97 been in development?

I started V96 from scratch in March 1996. By from scratch, I mean **from scratch**. I haven't used any code from 16-bit ViRC at all. V97 development began in March 1997, and although some of the V96 code was rewritten from the ground up, a lot of code was kept too.

How many lines of code does V97 have now?

V97 **1.00** consists of about **65000** lines of code.

Did you do all of this yourself?

Yes; there's no-one working on V97 except poor old me.

And the scripting engine?

I wrote the scripting engine all myself, from scratch.

Bugs in Borland's RTL?

Yep. Borland's RTL is known to contain bugs, although I've been able to work around many of them in V97.

Where does V97 store all its configuration information?

All information stored by V97 (including loaded scripts!) are placed in the registry under the key **HKEY_CURRENT_USER\Software\MeGALiTH Software\Visual IRC 96** (yes, that's **96**, to preserve compatibility with old ViRC '96 installations).

What does this DEFAULT.LIB file I have here do?

V97 is totally script-driven, and it doesn't know how to interpret and display any data from the server, nor does it have any built-in menus. DEFAULT.LIB is a small (currently around 14k) script which V97 loads only once, when you install for the first time. DEFAULT.LIB contains handlers for a large number of IRC events, installs some handy alias shortcuts, and installs some standard menus and popups. Once DEFAULT.LIB has been parsed and loaded into the registry after installation for the first time, it is not needed again.

Can I try TDCC Voice with you if I see you on EFnet #virc?

By all means.

Can I try TDCC Video with you if I see you on EFnet #virc?

No. I don't have a video camera myself, which is one of the reasons why there was such a delay between 0.82a and 0.91. It's unimaginably hard to code something that you can't really test yourself.

Is that all there is in this section at the moment?

Yes. Have a nice day.

About the author

ViRC '97 was written single-handedly by **Adrian Cable**.

I'm on **EFnet IRC** using the nick **MeGALiTH**, and I'm often to be found on **#quake**, when I'm not programming or killing my friends in E1M7. I'm also an IRC operator on **irc.ionet.net** and **irc.bofh.co.uk**. You can contact me via email at **acable@sv.span.com**, and the latest version of V97 can be found at **http://www.megalith.co.uk/virc**.

I'd appreciate that you report any bugs to me, **unless they are mentioned in this help file or on my web page**. I often get over 150 emails a day regarding V97, and many of those are "bug reports" telling me things that I've detailed on my page, or asking how they subscribe to the V97 mailing list ... or ... you get the picture. So only email if your problem is not listed elsewhere. I hope this doesn't sound too harsh - remember, the more time I have to spend replying to emails, the less time I'll have to actually code stuff. ;)

Finally - and this has nothing **whatsoever** to do with ViRC - if you have a C compiler anywhere (which you should do), cut & paste the lines below and compile and run them. It's a BASIC interpreter in 24 lines of C code! (I didn't write it, BTW). Prepare to be amazed. I was, and not a lot of things amaze me any more. It should work with any ANSI C compiler. I've tested it with **BCC** 4.0 and **GCC** 2.7.0.

This nifty program was written by **Diomidis Spinellis (dds@cc.ic.ac.uk)** for the 1990 International Obfuscated C Coding Contest. I apologise for failing to include the author's name in previous versions of this help file.

```
#define O(b,f,u,s,c,a)b(){int o=f();switch(*p++){X u:_ o s b();X c:_ o a b();default:p--;_ o;}}
#define t(e,d,_,C)X e:f=fopen(B+d,_);C;fclose(f)
#define U(y,z)while(p=Q(s,y))*p++=z,*p=' '
#define N for(i=0;i<11*R;i++)m[i]&&
#define I "%d %s\n",i,m[i]
#define X ;break;case
#define _ return
#define R 999
typedef char*A;int*C,E[R],L[R],M[R],P[R],l,i,j;char B[R],F[2];A m[12*R],malloc
(),p,q,x,y,z,s,d,f,fopen();A Q(s,o)A s,o;{for(x=s;*x;x++){for(y=x,z=o;*z&&*y== *z;y++)z+
+;if(z>o&&!*z)_ x;}_    0;}main(){m[11*R]="E";while(puts("Ok"),gets(B) )switch(*B)
{X'R':C=E;l=1;for(i=0;i<R;P[i++]=0);while(l){while(!(s=m[l]))l++;if (!Q(s,"\""))
{U("<>",'#');U("<=",'$');U(">=",'!');}d=B;while(*F=*s){*s=="&&j
++;if(j&1||!Q(" \t",F))*d++=*s;s++;}*d--=j=0;if(B[1]!='=')switch(*B){X'E':l=-1 X'R':B[2]!='M'&&(l=*--
C)X'I':B[1]=='N'?gets(p=B),P[*d]=S():(*(q=Q(B,"TH"))=0,p =B+2,S()&&(p=q+4,l=S()-
1))X'P':B[5]==""?*d=0,puts(B+6):(p=B+5,printf("%d\n",S ()))X'G':p=B+4,B[2]=='S'&&(*C++=l,p+
+),l=S()-1 X'F':*(q=Q(B,"TO"))=0;p=B+5;P[i =B[3]]=S();p=q+2;M[i]=S();L[i]=l X'N':+
+P[*d]<=M[*d]&&(l=L[*d]);}else p=B+2,P[ *B]=S();l++;}X'L':N printf(l)X'N':N free(m[i]),m[i]=0
        X'B':_ 0 t('S',5,"w",N fprintf(f,l))t('O',4,"r",while(fgets(B,R,f))(*Q(B,"\n")=0,G()))X
0:default:G()
;}_ 0;}G(){l=atoi(B);m[l]&&free(m[l]);(p=Q(B," "))?strcpy(m[l]=malloc(strlen(p )),p+1):
(m[l]=0,0);}O(S,J,'=',==,'#',!=)O(J,K,'<',<,'>',>)O(K,V,'$',<=,'!',>=) O(V,W,'+',
+,'-',-)O(W,Y,'*',*,'/',/)Y(){int o;_*p=='-'?p++,-Y():*p>='0'&&*p<= '9'?strtol(p,&p,0):*p=='('?p+
+,o=S(),p++,o:P[*p++];}
```

You can test it out with the following program, a text-based lunar lander thing. After running the interpreter, just type **OLD LANDER.BAS** and then **RUN**.

```
10 REM Lunar Lander
20 REM By Diomidis Spinellis
30 PRINT "You aboard the Lunar Lander about to leave the spacecraft." 60 GOSUB 4000
70 GOSUB 1000
80 GOSUB 2000
90 GOSUB 3000
100 H = H - V
110 V = ((V + G) * 10 - U * 2) / 10
120 F = F - U
130 IF H > 0 THEN 80
135 H = 0
140 GOSUB 2000
150 IF V > 5 THEN 200
160 PRINT "Congratulations!   This was a very good landing."
170 GOSUB 5000
180 GOTO 10
200 PRINT "You have crashed."
210 GOTO 170
1000 REM Initialise
1010 V = 70
1020 F = 500
1030 H = 1000
1040 G = 2
1050 RETURN
2000 REM Print values
2010 PRINT "           Meter readings"
2015 PRINT "           --------------"
2020 PRINT "Fuel (gal):"
2030 PRINT F
2040 GOSUB 2100 + 100 * (H <> 0)
2050 PRINT V
2060 PRINT "Height (m):"
2070 PRINT H
2080 RETURN
2100 PRINT "Landing velocity (m/sec):"
2110 RETURN
2200 PRINT "Velocity (m/sec):"
2210 RETURN
3000 REM User input
3005 IF F = 0 THEN 3070
3010 PRINT "How much fuel will you use?"
3020 INPUT U
3025 IF U < 0 THEN 3090
3030 IF U <= F THEN 3060
3040 PRINT "Sorry, you have not got that much fuel!"
3050 GOTO 3010
3060 RETURN
```

```
3070 U = 0
3080 RETURN
3090 PRINT "No cheating please!    Fuel must be >= 0."
3100 GOTO 3010
4000 REM Detachment
4005 PRINT "Ready for detachment"
4007 PRINT "-- COUNTDOWN --"
4010 FOR I = 1 TO 11
4020     PRINT 11 - I
4025     GOSUB 4500
4030 NEXT I
4035 PRINT "You have left the spacecraft."
4037 PRINT "Try to land with velocity less than 5 m/sec."
4040 RETURN
4500 REM Delay
4510 FOR J = 1 TO 500
4520 NEXT J
4530 RETURN
5000 PRINT "Do you want to play again? (0 = no, 1 = yes)"
5010 INPUT Y
5020 IF Y = 0 THEN 5040
5030 RETURN
5040 PRINT "Have a nice day."
```

Well, that's all from me ... enjoy.

Registration? No need ... it's freeware ... but ...

Unlike some other well-known IRC clients, ViRC '97 is **freeware**. That means there's no need to pay anything to use it. You can use it for as long as you like, and give it to your friends. However, feel free to send me a donation if you wish to show some appreciation for my efforts. V97 has been under development for around **1500 hours**. Even if you only send **$10**, you are effectively paying me less than **1 cent an hour**, which is **20000** times less than the salary of an experienced **American brain surgeon**. My address is as follows:

Adrian Cable
25 Halland Way
Northwood
Middlesex
HA6 2BY
United Kingdom

I accept payments in the form of cash, free software, free Internet accounts, and suggestions for what to include in future versions.

I must stress that sending me a donation entitles you to **nothing more** than a good feeling of warmth all over your body. I offer technical support equally to **anyone**, no matter how many millions of dollars they send me, unlike another IRC client, which only offers support to registered users. So, there's no need to pay me a cent, and no reason to either - unless **you** want to.

Thanks and acknowledgments

There are a few individuals who have done more than their fair share of testing, bug reporting, and generally wasting all their time on beta versions of V97 which didn't work properly, who I believe deserve a mention here. ;) They are (in alphabetical order):

Alan Miller - for finding a severe resource leak!!
Amagosa - for making **virc.com** actually happen and being great about everything else too.
Bort Vern - for spending hours each day testing out pre-alphas of my video conferencing code, which was made even more difficult by the fact that I don't have a cam myself.
DocWilco - for supporting ViRC right from the start, and still supporting it now.
FoX - for finding bugs in my menu/popup code that no-one else probably would have done.
][mm0rtal - for finally getting the video conferencing code to work fully.
JadeStar - for spending hours and hours finding the most obscure of bugs.
JasonG and **all the #virc ops** - for countless suggestions and bug reports.
Mr2001 - for showing me how to **really** write a hashtable algorithm.
Nailz - for telling me that the **$channels()** and **$channelcount()** functions documented in VSCRIPT.TXT didn't actually exist ;) (they're in 0.60, though). Also for giving me hints on optimizing ViRCScript, and inspiring me to make **TBestScroller**, the text output control that I wrote for use with ViRC '97 (wherever you see text, you're now seeing a **TBestScroller**).
ThinIce - for testing the real-time audio code in every pre-beta of 0.60, and for a number of useful suggestions for 0.62.

Thanks also go to **Yaron Gur** for a ton of suggestions, and to **Zenkevich Yury** for writing a better client than mine, and also for helping with the original real-time audio code in C++ that would have appeared in the never-to-be-released 16-bit 0.90alpha5.

I'm sure there are others too who I've forgotten here; you know who you are.

Finally, thanks go to **you** for bearing with V97 through many buggy alpha and beta releases. **You** make V97 what it is today.

Command-line parameters

Five **command-line parameters** are supported in **0.92**. They are described as follows. You can use as many command-line parameters together as you like.

-user *name*

Enables multi-user mode. This allows many users to use the same copy of ViRC '97, and store their own settings separately from the other users. For example, say that you had 3 users on your system, Jack, Jill, and Joe. Jack could run V97 with the parameter **-user jack**, Jill could run with **-user jill**, and Joe could run as **-user joe**. Then each of the 3 users are automatically given a separate key in the registry to store their own configurations, and no user will be able to affect any other user's settings. This works pretty transparently.

-exceptions

Crash-proof mode is now enabled by **default**. This means that if you see odd behaviour and want to show any **exceptions** that occur, and then report them to me, rerun V97 with the **-exceptions** parameter. I'm not sure how well this will work yet - I'd appreciate reports of success/failure stories while using/not using this option.

-noaudio

If you're having problems starting up (for example, V97 displays the button on the taskbar but never loads fully), it could be because the audio initialization code is causing your machine to lock up. If this is the case, running V97 with the **-noaudio** parameter will disable real-time audio and should enable everything else to work correctly.

-novideo

Prevents V97 from trying to initialize your video capture hardware. If you don't have a video capture card, or don't wish to appear to other users who want to video conference with you, you can run V97 with the **-novideo** parameter to surpress support for outgoing video.

-debug

Enables debugging mode, and the **Debug** menu, which can be very useful for debugging scripts. The items on the **Debug** menu perform the following functions:

Process log

Enables logging of all server data received, and every ViRCScript command and function executed, along with what it evaluates to, amongst other information, to the file **DEBUG.LOG**. Very useful for debugging scripts.

Script parser ...

Enables you to type fragments of ViRCScript code for immediate execution.

Expression evaluator ...

Lets you enter ViRCScript expressions for immediate evaluation. Great for testing your own functions.

Write ViRCScript library

Writes all your events and aliases out to **DEFAULT.LIB**. Remember to backup your old DEFAULT.LIB before using this function!! (This is pretty useless now, though, as the ViRCScript Publishing Wizard does this in a far better way).

Dump profiler table

This function does nothing in all release versions of V97. In internal versions, it writes out the current profiler table (indicating performance of various internal functions) to the debug log.

Memory usage stats

Pops up a window showing the status of the memory manager. See how much memory V97 is hogging. =]

Adding ViRCScript scripts to your desktop

When you install V97 for the first time, a number of file associations are registered with the Windows 95/NT shell. V97 is set as the viewer for **ViRCScript** files (.vsc) and **Global Chat** files (.chat).

To add a ViRCScript script to your Windows 95/NT desktop, right-click on a blank area of the desktop, go down to New, and select **ViRCScript Script File**. Then type a name for your script. Right-clicking on the ViRCScript file will bring up a context menu, including the items **Execute script** and **Edit script**. Clicking on **Execute script** (or double-clicking the icon) will load up V97 and load and execute the script. Clicking on **Edit script** will load the script up in **Notepad** for editing. Note that you can't currently alter the application used for script editing - this will come later (probably in the next release).

Adding IRC server connections to your desktop

When you install V97 for the first time, a number of file associations are registered with the Windows 95/NT shell. V97 is set as the viewer for **ViRCScript** files (.vsc) and **Global Chat** files (.chat).

To add an IRC server connection to your Windows 95/NT desktop, right-click on a blank area of the desktop and select **ViRC '97 IRC Server Connection** from the **New** popup menu. Then right-click on the icon. Two of the options available in the context menu are **Connect to IRC** and **Edit server connection**. Clicking on **Connect to IRC** (or double-clicking on the icon) will attempt to connect to IRC using that server configuration. However, you have yet to define a server configuration, so an **edit desktop server connection** dialog box will appear. Here, you can enter a **title** for the connection, the **server** and **port** you wish to use, and, optionally, a **channel** to join when you have connected. Then press OK to save the settings to the file and join the server. Next time you want to connect to that server configuration, simply double-click on the icon. If you want to change the details of that connection, right-click on the icon and select **Edit server connection**.

V97's server connection files are created in **.chat** format, which can be read by Quaterdeck's Global Chat, amongst other IRC clients. In addition, you can run **.chat** files from anywhere you want to load them up in V97 as V97 creates an association for **.chat** files.

ViRCScript Publishing Wizard

The **ViRCScript Publishing Wizard** is a powerful new tool available in **V97 0.63** and later versions which allows you to generate distributable ViRCScript **.vsc** files from aliases, events and menus/popups that you've defined from within V97. You can go into the wizard by selecting **ViRCScript Publishing Wizard ...** from the **Scripting** menu.

The wizard consists of a number of **pages**. Click **Next** to go onto the next page, **Back** to return to the previous page to change some options, **Abort** to cancel the wizard and **Finish** to complete the wizard once all the pages have been filled in. The different pages are listed below.

Start

Enter the name and version number of your script here.

Aliases

Select the aliases you wish to include in the script here. You may press **All** to select every alias for inclusion, and **None** to deselect all aliases.

Events

Select the events you wish to include in the script here. You may press **All** to select every event for inclusion, and **None** to deselect all events.

Menus

Select the menus and popups you wish to include in the script here. You may press **All** to select every menu and popup for inclusion, and **None** to deselect all menus and popups.

Finish

Your script is now ready to be generated. There is one option here. Select the checkbox to **reflow and indent** the script. This formats the script nicely, indenting code blocks and making it easier to read, and is strongly recommended so other people can understand your script.

Once the wizard is complete, a **.vsc** file will be generated containing your complete script. Other users may then use the **/load** command to load your script.

Client setup - MDI (child) window settings

In this tab of **client setup**, you can select which **MDI child windows** you wish to appear on the desktop. Desktop windows are free from the confines of the main V97 window, and can be moved anywhere you wish.

Tasktray watchdog

New in V97 **0.80** and above is the **tasktray watchdog**. To enable the watchdog, simply press the **V97** logo toolbar button on the main toolbar.

When the **tasktray watchdog** is enabled, V97 will almost completely vanish, leaving only a small V97 icon on the system tray (next to where the clock is in Windows 95). However, if you receive a private message, a message box will pop up with the sender and the message, and will ask if you want to bring V97 back up. If you click **Yes**, the watchdog icon will **disappear** and the main V97 window will come back up. If you click **No**, V97 will stay in watchdog mode. If you want, you can force V97 to leave watchdog mode by **double-clicking** on the tasktray icon. If you hold the mouse over the icon for a few moments, a **tooltip** will appear, informing you if you have any new messages waiting while the watchdog was active.

Client setup - Chat logging

In this tab of **Client setup**, you can control what sort of things V97 automatically **logs** to a file.

Server logging options

Pressing the **Enable server logging** checkbox will automatically make V97 log any miscellaneous things received from the server that doesn't fit into any other logging category to a file, which is set in the **Server log file** edit box. The default file is **server.log**.

Channel logging options

Pressing the **Enable channel logging** checkbox will automatically make V97 log anything displayed in a channel window to a file, which is set in the **Channel log file** edit box. The default file is **$channel.log**, which means, for example, that everything in the channel **#quake** will be logged to the file **#quake.log**.

Query logging options

Pressing the **Enable query logging** checkbox will automatically make V97 log anything displayed in a query (private message) window to a file, which is set in the **Query log file** edit box. The default file is **$nick.log**, which means, for example, that a query conversation with **MeGALiTH** will be logged to the file **MeGALiTH.log**.

DCC Chat logging options

Pressing the **Enable DCC Chat logging** checkbox will automatically make V97 log anything displayed in a DCC Chat window to a file, which is set in the **DCC Chat log file** edit box. The default file is **$nick.dcc.log**, which means, for example, that a DCC Chat conversation with **MeGALiTH** will be logged to the file **MeGALiTH.dcc.log**.

Client setup - Double-click actions

In this tab of **Client setup**, you can control what IRC commands are executed when the user **double-clicks** on different parts of different windows.

Server text

The IRC command entered here is executed whenever the user **double-clicks** on the **server text** pane of a **server notices window**. This is blank by default, but you can set it, for example, to **/connect**, which will connect to the server when you double-click on the window.

Channel text

The IRC command entered here is executed whenever the user **double-clicks** on the **channel text** pane of a **channel window**. This is blank by default, but you can set it, for example, to **/wwho**, which will show a detailed listing of all the users on the channel when you double-click on the window.

Channel nicks

The IRC command entered here is executed whenever the user **double-clicks** on a **nickname** in the **nicks pane** pane of a **channel window**. This is set to **/whois $1** by default, which brings up a **WHOIS** dialog for that user whenever you click on a nickname. If, for example, you'd rather have it bring up a **query (private message) window** for that user, you can, for example, set it to **/query $1**.

Client setup - Text output settings

In this tab of **Client setup**, you can adjust the **text output settings** that control V97's text windows.

Tab stop settings

Here, you can change the number of spaces that a tab represents. This affects the spacing between the nickname and the spoken text in channel messages, and so forth, and a sample of what text looks like with the specified tab setting is shown.

Text box settings

This setting controls how many lines ViRC '97 will scroll at once. The recommended setting is the default - 25 lines. This gives fast text output that shouldn't be too jerky. Setting this to the maximum - 50 lines - will result in lightning-fast text output, but if a lot of channel activity occurs it may scroll off the screen in one go before you've even read it!! If you have a very fast PC, you might want to try lowering this setting, say, to 5 or 10 lines, which results in smoother text scrolling.

Disable info bitmaps in text boxes

Do you like the little arrows and things that display to the left of the text when people join channels and so forth? If not, select this checkbox to disable them.

Stamp every message displayed with the current time

This option does exactly what you'd expect - every line of text displayed anywhere is prefixed with the current time.

Time stamp format

In this box, you can change the format of the timestamp that's displayed. Time formats are described in detail in **VSCRIPT.TXT** under the **[2.5.6.8] Time function** section, but, to sum up, **hh** represents hours, **nn** represents minutes, (yes, that's **nn**. **mm** represents the month), and **ss** represents seconds. **\b** attributes are used to display the time stamp string in bold (**\u**, **\i** or **\k** can also be used if desired - see **VSCRIPT.TXT** for more information on attributes).

Client setup - Local IP address

In this tab of **Client setup**, you can adjust the **IP address** that V97 uses for outgoing **DCC** connections, or you can leave it to grab your IP **automatically**, which is the default setting.

If outgoing DCCs are not working, and you have a **fixed IP address**, if your PC has **multiple IP addresses** (for example if you have an ethernet card running TCP/IP and you connect to the Internet via a dial-up) and V97 is grabbing the wrong one for use with DCC, you can enter your real IP address here and V97 will use this instead.

Outgoing DCCs will **not work** if the IP address is entered incorrectly here, which is why it's a good idea generally to leave the **Automatic configuration** checkbox enabled unless you have any of the problems described above.

What do you if you have problems

In this section, I'll attempt to detail a number of common problems (along with their solutions), and the procedure to follow if you want your own problems answered, or if you wish to report a bug.

- **Q: ViRC '97 installs fine, but when I connect to a server and join a channel I cannot see what anyone else is saying, but I can see what I type myself.**
  A: For some reason, on a few systems, the default script library, DEFAULT.LIB, is not loaded correctly. The solution is very simple. In a server window, simply type:

  /load c:\virc97\default.lib

  (replace c:\virc97 with the path that you have ViRC '97 installed in).

- **Q: Real-time voice chat (TDCC Voice) does not work on my machine. When I started ViRC '97 up for the first time, I got an error stating that my audio hardware could not be initialized.**
  A: On some machines, the GSM voice compression driver is not set up correctly when Windows 95 or NT is installed. The way round this is as follows. Go into Control Panel and go into the Multimedia section. Then go into the Advanced tab and double-click on Audio Compression Codecs. Double-click on Microsoft GSM 6.10 Audio CODEC, and click on the Settings... button. Then make sure Compression and Decompression are both set to All rates. This doesn't work on every system, although many people have reported success after carrying out this procedure.

- **Q: Real-time video conferencing (TDCC Video) does not work on my machine. In the Video conferencing tab in Client setup, I get a message telling me my video capture hardware could not be initialized.**
  A: First of all, video capture hardware is not necessary to video conference with others, although of course you cannot send them video. If you do actually have video capture hardware, check that you have the latest drivers. The Connectix QuickCam is a very popular camera which works very well with TDCC Video, but the older drivers for it may fail. You can download the latest QuickCam drivers from http://www.connectix.com, and these work fine.

It's inevitable that some of you will:

- Have a problem which is not covered here
- Have difficulty writing a script
- Encounter a bug

If so, your first course of action should be to join the **EFnet** channel **#virc**. There are always people here who are willing to (attempt to) answer any questions you may have, ranging from problems with the basic operation of the client to advanced scripting questions. If you have a bug to report, the people on **#virc** will report the problem to me, and it should be fixed for the next version.

If the people on **#virc** are **unable to help with your problem**, then, **and only then** should you email me at **acable@sv.span.com**.

I frequently get over **150** emails a day regarding ViRC '97. Many of them concern the same bug, or are sent by many different people who all have the same difficulty. I am backlogged by at least a week, and, to be honest, it is often difficult to cope with the immense volume of email, given that I have a full-time job, and occasionally need a break from programming and answering emails when I get back in the evenings. Nonetheless, I try my best, and attempt to reply to every email I receive. I really hope that I will continue to be able to do this, and the best way to ensure this is to only email me as a last resort.

I hope you all understand, and I hope this doesn't sound too harsh!! :)

Using text attributes and colour

In all ViRC '97 windows, **attribute** codes and **colour** codes may be entered. The attribute codes will appear as **blocks** when they're entered, but will display correctly when they appear in the text output area. The following attribute codes are currently supported:

- **Ctrl+B** - makes text **bold**
- **Ctrl+U** - makes text {/u underlined /u}
- **Ctrl+N** - makes text *italic*
- **Ctrl+O** - turns off all **bold**, {/u underlined /u} and *italic* attributes.

In addition, **Ctrl+K** can be used to enter **colours**. To select a colour, press **Ctrl+K** followed by a number from **1** to **15**. The following colours are available:

0 - white
1 - black
2 - blue
3 - green
4 - red
5 - maroon
6 - purple
7 - orange
8 - yellow
9 - bright green
10 - cyan
11 - bright blue
12 - light blue
13 - light purple
14 - dark grey
15 - light grey

A background colour for text can also be specified by putting a comma after the foreground colour number, and then putting the background colour number, e.g. **Ctrl+K** followed by **1,9** would give black text on a bright green background.

Client setup - Video conferencing

In this tab of **Client setup**, you can test your video capture hardware and change the settings.

Press the **Record** button to start displaying video in the panel, and **Stop** to stop. This way, you can check that the video appears correctly, and the camera is in focus. You can also adjust the **Capture rate** slider which determines how many frames a second V97 will attempt to send to your other party.

It is worth mentioning that, unless you have a fast link (T1 or above), the **Capture rate** slider should be set very low (e.g. to 2 fps), as increasing it will just increase CPU usage without resulting in any improvement, as the limiting factor will be the speed of your link here.

Client setup - Voice chat

In this tab of **Client setup**, you can change the **buffer size** used by TDCC Voice.

The default buffer size is **18 blocks**. If you are frequently getting broken up voice, even though you are on a good connection (28.8 kbit/sec modem or above), you should increase the buffer size. Note that increasing the buffer size will also increase **latency** (the delay between when the other user speaks and when you hear what they say), so setting it too high is a very bad idea. If you and the other user are both on very fast links (T1 or above), you can set the buffer size to **1** or **2 blocks** for very low latency without any risk of break-up.

Client setup - Download extensions

In this tab of **Client setup**, you can associate actions to files of a particular extension when they're downloaded.

ViRC '97 comes with a number of extensions defined by default. You should find the existing extensions serve as useful examples if you ever need to define your own.

Add

Adds an extension to the list.

Remove

Removes an extension from the list.

Save to dir

If you'd like files of this extension to be saved in a directory different from the download directory you have specified in the **Paths** tab of **Client setup**, press the **...** button and select the directory.

Command

Enter here an IRC command to handle the file of this extension. For example, to handle WAV and MID files, you could enter **/mci play $file** here, and to handle VSC (ViRCScript) files, you could enter **/load "$file"** here.

Automatically execute when file downloaded

When a file of this extension is downloaded, automatically execute what's in the **Command** field when the download is complete.

Client setup - Scripts (custom)

This tab in **client setup** normally contains a screen detailing the number of aliases, events, objects and classes installed, and also the version and build numbers of ViRC '97 and the OVS interpreter.

However, information is given on this page how to write scripts which add pages to this tab, which can contain your own custom script options that the user can set.

Client setup - Active Scripting

Active Scripting is a powerful new feature of ViRC '97 which allows custom language DLLs to be loaded in, which can then be used to write scripts in (see **AXSCRIPT.TXT** for more details on how to use Active Scripting languages in ViRCScript code).

This tab of **client setup** allows you to install new Active Scripting DLLs, or to detect any that are already installed for use with ViRC '97.

Microsoft Internet Explorer 3.0 and above come with VBScript and JavaScript DLLs. They are installed when you install IE itself, so they should appear in this list. If you donwload any additional language DLLs that don't have automatic installation, you may have to install them manually, as follows.

Click on the **Install language ...** button to register and install a new language DLL on the system. This language DLL will then become available for use by any application that supports Active Scripting, including Microsoft Internet Explorer and (of course) ViRC '97.

Click on **Rebuild installed language list** to automatically add any language DLLs installed to the list. Once in the list, these languages can be used from ViRCScript code. If you have just installed a new language DLL from outside of ViRC '97, you will need to press this button for ViRC '97 to "find" the language so it can be used from ViRCScript code.

Client setup - Colours and fonts

In this tab of **client setup**, you can change how ViRC '97 displays text visually by using different colours and fonts for different types of text.

Screen areas

In this list box, you can select from a large number of different screen areas. Screen areas ending in **output pane** are the actual text area where new text appears. The background colour of this, and the font that it's displayed in, can be changed from here. To change the foreground colours of text, you need to select an area such as **Channel window: regular text**. Note that the fonts of each type of text cannot be changed individually - to change the font for all types of text in that particular area, select, for example, **Channel window: output pane** and change the font from there.

Colour

From here, a foreground colour for an area (if applicable) can be selected by left-clicking on a colour, and a background colour for an area (if applicable) can be selected by right-clicking on a colour. Custom colours that don't appear in the colour square can also be used by pressing the **FG ...** or **BG ...** buttons to bring up the custom colour selector.

Select from font list ...

Pressing this button changes the font (if applicable) for the individual text area selected.

Client setup - Flood protection

In this tab of **client setup**, you can change ViRC '97's flood protection options. Floods occur when something causes ViRC '97 to send out too many lines to the server within a short period of time. One thing that can cause this is a malicious attack by someone else, so it's important to be protected from floods.

Enable flood protection

Selecting this checkbox enables flood protection. It is enabled by default, and it is recommended that it be left on at all times. You can change the flood protection level by moving the slider. The slider controls the maximum number of lines that can be sent per second. If you try to send more lines, they will be buffered, and sent as soon as possible afterwards, but still ensuring that this limit is never exceeded.

Move the slider to the left to **INCREASE** the protection, although this will also increase latency (the delay between command submission and processing). Move the slider to the right to **DECREASE** the protection, as this decreases latency too. The default value of 5 should be good enough for most purposes - it offers good protection, yet the latency added is unnoticable.