Visual IRC '96 Help

Welcome to **Visual IRC '96** version **0.60**, the leading freeware script-driven IRC (Internet Relay Chat) client for Microsoft Windows 95 and Windows NT with real-time audio chatting (see the **DCC commands** section).

This help file is not yet complete. There may be certain aspects of V96 that are not covered here. If so, I apologize, and will endeavour to add them to a future version of this help file.

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

Introduction and installation

Starting your IRC session

Joining an IRC channel

IRC commands

Client setup

http://apollo3.com/~acable/virc.html

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 '96**, 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 '96 above all other IRC clients) at **http://www.cwsapps.com/95irc.html**.

Installation

ViRC '96'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
- Windows **95** or Windows **NT** (**3.51** or **4.0**)
- **8mb** RAM (more needed for NT)
- Just over **1mb** of hard disk space

Quick installation:

- Make a **new directory** on your hard disk, calling it **VIRC96**.
- Unzip the V96 distribution ZIP (called **virc96_*xxx*.zip**, where ***xxx*** is the version number) into that directory, using WinZIP or DOS pkunzip.
- Run **VIRC96.EXE** to begin setup.

Setting ViRC '96 up if you already use mIRC, WSIRC, PIRCH, or 16-bit ViRC:

- Click on **I use mIRC, WSIRC, PIRCH, or ViRC16** 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 will be imported automatically. ViRC '96 will then display the standard **User Setup** dialog to make sure your settings are correct.

Setting ViRC '96 up for the first time:

- Select **I want to start a fresh installation** and click **Install**.
- 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 **replacing** the final character with a **_**.
- 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 '96 will be ready for use.

Why use ViRC '96?

Why use ViRC '96 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 '96 brings you the **best of both worlds** - and real-time audio chatting 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 '96 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 '96 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 '96
- **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 '96 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 '96!!

Server window

Each server window has a **toolbar**. If you wish to conserve screen space, you may turn it off by clicking on the **red magic dot**. To turn it back on, click on the **yellow magic dot**. The toolbar buttons, from left to right, are as follows:

- Connect to server
- Disconnect from server
- Show channel box
- Show server links
- Show channel list
- Print

The toolbar will be **customizable** in future versions of ViRC '96.

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.

Channel box

ViRC '96 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 two **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**

ViRC '96 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.

Channel window

Each channel window has a **toolbar**, similar to that of the server window.. If you wish to conserve screen space, you may turn it off by clicking on the **red magic dot**. To turn it back on, click on the **yellow magic dot**. The toolbar buttons, from left to right, are as follows:

- Leave channel
- Cycle channel (leave and rejoin)
- Print

The toolbar will be **customizable** in future versions of ViRC '96.

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

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 '96) 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
```

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 '96 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 '96 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** and **send**. Current supported TDCC types, specific to V96, are **send** and **voice**, although **whiteboard** will be supported in a future release.

DCC CHAT command

Usage: /dcc chat *nick*

Initiates a direct, secure **chat** session with *nick*.

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.

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 or IaIRC to receive files by TDCC. If *file* is omitted, you will be prompted to select one or more files to send.

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.

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 you may speak and listen to the other user at the same time.

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.

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. V96 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 '96'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 V96 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 V96 receives a **SOUND** request, V96 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 V96.

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 V96 (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.

Client setup - User 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, V96 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 V96's **DCC** support.

Blocksize

Files are transferred block-by-block by V96. 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 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 connections

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 Send connections

Similar to the above, only automatically accepts **DCC Send** connections. Enable this option with **caution**, as files that already exist in your V96 **download directory** will be overwritten without confirmation if this option is enabled.

Automatically minimize DCC Chat window

Whenever a DCC Chat window appears on the screen, makes V96 discreetly minimize it to cause the minimum possible disruption.

Automatically minimize DCC Send window

Whenever a DCC Send/Get window appears on the screen, makes V96 discreetly minimize it to cause the minimum possible disruption.

Client setup - Hyperlinks

In this tab of **client setup**, you can configure how V96 interfaces with your **web browser**. V96 comes preconfigured with settings for **Netscape**, **Enhanced Mosaic** and (soon) **Internet Explorer**, so all you'll have to do is double-click on your browser to set it as the default.

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**. V96 replaces the **&** in the item field with the URL to send before hyperlinking.

Client setup - IDENTD

This tab in **client setup** allows you to turn on and off V96's built-in **IDENTD** server. 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.

Enable IDENTD server

Turns on V96's built-in IDENTD server.

Automatic configuration

This will make V96 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 V96'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 V96'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.

Client setup - Finger server

This tab in **client setup** allows you to configure V96's built-in **finger server**. Most UNIX systems provide a built-in finger server, which allow other users to query information about you, and V96 offers the same sort of feature.

Enable finger server

Turns on V96's built-in finger server. V96 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 V96'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 V96.

Enable SOCKS support

Tells ViRC '96 to connect to IRC servers through a SOCKS server. Note that DCC connections are not yet supported.

Automatic configuration

Enabling this option simply instructs V96 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 V96's **XDCC** support. XDCC provides an automated way for people to query and request files from you, much like a file server.

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.

Client setup - Userlist

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.

V96 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 V96 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 V96 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 V96 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

This tab in **client setup** allows you to configure V96'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

In this tab of **client setup**, you can specify a list of nicknames. V96 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 V96 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 - Windows

In this tab of **client setup**, you can adjust how V96 displays its windows.

Entry box height adjust

You can adjust the height of the entry box in V96 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 V96'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 V96 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.

Client setup - Uninstall

Pressing the **Uninstall** button in this tab of **client setup** causes V96 to prepare itself for uninstallation. It will remove everything it has added to your registry, and exit. It will not remove any files from your hard disk. In order to complete the uninstallation, you must drag the ViRC '96 directory into the recycle bin using Windows Explorer or delete the directory from an MS-DOS box using the **DELTREE** command.

Client setup - Paths

In this tab of **client setup**, you can change where V96 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 - Options

In this tab of **client setup**, you can change a number of **options** that control the way ViRC '96 works. In this tab are 3 **subtabs**, which are detailed below.

***Server options* tab**

This tab contains the following server-related options.

Auto-connect to server on startup

Makes V96 open a new connection window and try to connect to the default server automatically on startup.

Auto-reconnect to server if connection failed

Makes V96 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 V96 cannot connect to the server, for example, if the connection is refused, V96 will not try to reconnect, regardless of this setting.

Retry to connect ... times

Sets the number of times V96 will try to connect before giving up.

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.

***Aesthetic options* tab**

This tab contains a number of options relating to the way V96 displays things.

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.

Save window positions and sizes on exit

Makes V96 save the position and sizes of the main window when the user quits. In a future release, all the window positions will be saved, not just those of the main 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 V96 from sending a pong back to the server, it'll just stop the display.

Enable button hover-highlighting

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

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, it should be left **on**. System performance is no longer degraded by this option.

***Miscellaneous* tab**

Here you can set miscellaneous settings which don't really fit into the other 2 groups.

Keep channel box and channel list open after joining a channel

Enabling this option prevents V96 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 V96 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.

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.

Client setup - Defaults

In this tab of **client setup**, you can set a number of default strings and options that V96 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 V96's title bar. It's safe to leave this alone (unless you want to change the format, of course).

Scripting - Aliases

To edit V96's **aliases**, select **Aliases...** from the **Scripting menu**.

This section does not detail **ViRCScript**, the language used to write aliases in V96. 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 V96 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.

V96 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 - V96 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 V96 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 $C Hello everyone!!
```

I've had people writing to me asking why their **Say** aliases don't work. 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 V96, not mIRC or PIRCH. >:->

Scripting - Events

To edit V96'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 V96 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 V96. 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 V96 ZIP.

What is an event?

Unlike every other IRC client, even ircII, V96 is totally **event-driven**. What this basically means is that V96 by itself has no idea how to interpret any data received from the IRC server. All text displayed by V96 is actually displayed by **events**, which are fired by V96 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 V96 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. V96 will handle this automatically.

A complex priniciple to grasp at first is event **priority**. Basically, for each line of server text received, V96 will fire only **one** event. If the line received matches the mask of more than one event, the one with the highest **priority** will be fired. For an example of where priority is needed, compare the **ChannelMessage** event with the **PrivateMessage** event. Can you work out why **ChannelMessage** requires a higher priority to prevent all incoming channel messages from being displayed like private messages? An example channel message is:

> **:nick!user@host PRIVMSG #channel :hello all!!**

An example private message is:

:nick!user@host PRIVMSG mynick :hello!!

The mask for **PrivateMessage** is *** PRIVMSG ***. However, this event would also be fired if a channel message is received. In order for this to be prevented, the **ChannelMessage** mask must be *** PRIVMSG #*** and the priority set higher to prevent channel messages from being handled by the **PrivateMessage** event. You **will** get used to this when writing your own scripts, it's just a tough concept to get right at first. Early versions of V96's default event library, **DEFAULT.LIB**, had some priorities set wrongly which meant that certain CTCPs were displayed wrongly. If I can get it wrong, you can too ... but with experience you'll be able to do it perfectly.

V96 displays a little ball beside each event. The ball is normally green, however, if you have modified the events that session, the ball will go red. This allows you to see at a glance which events you have tampered with if you're fiddling with events and something doesn't work any more.

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 **priority** (see above) and the **mask**. If you want to make a new event for the server code **353**, for example, you could enter **ServerEvent353** for the name, **5** for the priority, 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.

Priority

Enter the **priority** for the event in this field.

Mask

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

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 anything about it to start coding events. See **VSCRIPT.TXT** for more information on coding events.

Scripting - Menus/popups

To edit V96'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 V96. 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 V96 ZIP.

Introduction

Unlike every other IRC client, V96 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 V96 when you install it for the first time.

Menus and popups in V96 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 V96, menus are defined in a similar way. When you define an item, you have to give it a depth, whic his the depth of the item in the tree. In the examples above, **File** and **Edit** would have a depth of **0**, as they are base items, and all the subitems would have a depth of **1**. 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**. So the concepts behind menus are very, very simple as you can see (currently, V96 supports a depth of **0**, **1** and **2** only. Future versions may support greater depths).

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	- V96's main window menu bar
Server popup	- right-click menu for server window
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 popup

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 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**.

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

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 '96

Lots of people ask things like what language V96 is written in, and so forth, so I'll try to answer the most common ones here.

What language is V96 written in?

V96 is written totally in Borland Delphi 2.0 Developer.

How long has V96 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.

How many lines of code does V96 have now?

V96 0.60 has no more, and no less, than **24457** lines of code.

Did you do all of this yourself?

Yes; there's no-one working on V96 except poor old me.

What about custom controls? And the scripting engine?

I wrote the scripting engine all myself, from scratch. The only custom control I use is a sockets control. And I wrote that, too. So all problems you encounter are either true bugs in V96 or bugs in Borland's RTL.

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 V96.

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

V96 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 8k) script which V96 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 accessed again.

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

Yes. Have a nice day.

About the author

ViRC '96 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. You can contact me via email at **acable@sv.span.com**, and the latest version of V96 can be found at **http://apollo3.com/~acable/virc.html.**

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 100 emails a day regarding V96, and many of those are "bug reports" telling me things that I've detailed on my page, or asking how they subscribe to the V96 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.

```
#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],I,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;I=1;for(i=0;i<R;P[i++]=0);while(I){while(!(s=m[I]))I++;if (!Q(s,"\""))
{U("<>",'#');U("<=",'$');U(">=",'!');}d=B;while(*F=*s){*s=="&&j
++;if(j&1||!Q(" \t",F))*d++=*s;s++;}*d--=j=0;if(B[1]!='=')switch(*B){X'E':I=-1 X'R':B[2]!='M'&&(I=*--
C)X'I':B[1]=='N'?gets(p=B),P[*d]=S():(*(q=Q(B,"TH"))=0,p
=B+2,S()&&(p=q+4,I=S()-1))X'P':B[5]==""?*d=0,puts(B+6):(p=B+5,printf("%d\n",S
()))X'G':p=B+4,B[2]=='S'&&(*C++=I,p++),I=S()-1 X'F':*(q=Q(B,"TO"))=0;p=B+5;P[i
=B[3]]=S();p=q+2;M[i]=S();L[i]=I X'N':++P[*d]<=M[*d]&&(I=L[*d]);}else p=B+2,P[ *B]=S();I++;}
X'L':N printf(I)X'N':N free(m[i]),m[i]=0 X'B':_ 0 t('S',5,"w",N fprintf(f,I))t('O',4,"r",while(fgets(B,R,f))
(*Q(B,"\n")=0,G()))X 0:default:G()
;}_ 0;}G(){I=atoi(B);m[I]&&free(m[I]);(p=Q(B," "))?strcpy(m[I]=malloc(strlen(p )),p+1):(m[I]=0,0);}
O(S,J,'=',==,'#',!=)O(J,K,'<',<,'>',>)O(K,V,'$',<=,'!',>=) 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 '96 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. V96 has been under development for around **1000 hours**. Even if you only send **$10**, you are effectively paying me **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 V96 which didn't work properly, who I believe deserve a mention here. ;) They are:

JadeStar - for spending hours and hours finding the most obscure of bugs.
FoX - for finding bugs in my menu/popup code that no-one else probably would have done.
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.
ThinIce - for testing the real-time audio code in every pre-beta of 0.60.

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 V96 through many buggy alpha and beta releases. **You** make V96 what it is today.

Command-line parameters

Three command-line parameters are supported in 0.60. They are described as follows. Note that you can currently only use one parameter at a time - this will be "fixed" in a future version.

-user *name*

Enables multi-user mode. This allows many users to use the same copy of ViRC '96, 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 V96 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.

-crashproof

Enables crash-proof mode. This simply silences all exceptions that occur. If you encounter **access violation errors** frequently, you might want to run V96 with the **-crashproof** parameter and see if it works any better. **Please email me at acable@sv.span.com and let me know your experiences with crash-proof mode**.

-debug

Enables debugging mode, and the **Debug** menu. This is pretty useless now that I've removed most of the debug code in 0.60. The items on the **Debug** menu perform the following functions:

Process log

Enables logging of all server data received, amongst other information, to the file **DEBUG.LOG**.

Script parser

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

Write ViRCScript library

Writes all your events and aliases out to **DEFAULT.LIB**. Remember to backup your old DEFAULT.LIB before using this function!!

Dump profiler table

This function does nothing in all release versions of V96. In internal versions, it writes out the current profiler table (indicating performance of various internal functions) to the debug log.