GO-Global TN User Manual

GO-Global TN is a Windows Telnet and SSH client that can tunnel arbitrary data through a login session. This manual documents GO-Global TN.

Contents

Chapter 1: Introduction to GO-Global TN

GO-Global TN is a free SSH, Telnet and Rlogin client for 32-bit Windows systems. In conjunction with a host-based component, it can tunnel arbitrary data through a Telnet, Rlogin, or SSH session. It is designed for use with GO-Global UX to allow traversal of firewalls with advanced authentication methods, or other restrictive network topologies.

1.1 What are SSH, Telnet and Rlogin?

If you already know what SSH, Telnet and Rlogin are, you can safely skip on to the next section.

SSH, Telnet and Rlogin are three ways of doing the same thing: logging in to a multi-user computer from another computer, over a network.

Multi-user operating systems, such as Unix and VMS, usually present a command-line interface to the user, much like the ‘Command Prompt’ or ‘MS-DOS Prompt’ in Windows. The system prints a prompt, and you type commands which the system will obey.

Using this type of interface, there is no need for you to be sitting at the same machine you are typing commands to. The commands, and responses, can be sent over a network, so you can sit at one computer and give commands to another one, or even to more than one.

SSH, Telnet and Rlogin are *network protocols* that allow you to do this. On the computer you sit at, you run a *client*, which makes a network connection to the other computer (the *server*). The network connection carries your keystrokes and commands from the client to the server, and carries the server's responses back to you.

These protocols can also be used for other types of keyboard-based interactive session. In particular, there are a lot of bulletin boards, talker systems and MUDs (Multi-User Dungeons) which support access using Telnet. There are even a few that support SSH.

You might want to use SSH, Telnet or Rlogin if:

- you have an account on a Unix or VMS system which you want to be able to access from somewhere else
- your Internet Service Provider provides you with a login account on a web server. (This might also be known as a *shell account*. A *shell* is the program that runs on the server and interprets your commands for you.)
- you want to use a bulletin board system, talker or MUD which can be accessed using Telnet.

You probably do *not* want to use SSH, Telnet or Rlogin if:

- you only use Windows. Windows computers have their own ways of networking between themselves, and unless you are doing something fairly unusual, you will not need to use any of these remote login protocols.

1.2 How do SSH, Telnet and Rlogin differ?

This list summarises some of the differences between SSH, Telnet and Rlogin.

- SSH is a recently designed, high-security protocol. It uses strong cryptography to protect your connection against eavesdropping, hijacking and other attacks. Telnet and Rlogin are both older protocols offering minimal security.
- Telnet allows you to pass some settings on to the server, such as environment variables. (These control various aspects of the server's behaviour. You can usually set them by entering commands into the server once you're connected, but it's easier to have Telnet do it automatically.) SSH and Rlogin do not support this. However, most modern Telnet servers don't allow it either, because it has been a constant source of security problems.
- SSH and Rlogin both allow you to log in to the server without having to type a password. (Rlogin's method of doing this is insecure, and can allow an attacker to access your account on the server. SSH's method is much more secure, and typically breaking the security requires the attacker to have gained access to your actual client machine.)
- SSH allows you to connect to the server and automatically send a command, so that the server will run that command and then disconnect. So you can use it in automated processing.

The Internet is a hostile environment and security is everybody's responsibility. If you are connecting across the open Internet, then we recommend you use SSH. If the server you want to connect to doesn't support SSH, persuade the administrator to install it.

If you are behind a good firewall, it is more likely to be safe to use Telnet or Rlogin, but we still recommend you use SSH.

1.3 How does GO-Global TN tunnel data through firewalls?

GO-Global TN uses a host-based component to multiplex data streams on the Telnet, SSH, or Rlogin connection. The data is received on the Windows machine, transmitted using a proprietary protocol to the UNIX host, and then re-sent by the host component to a service on the remote network.

The benefits of this method are:

- Some sites restrict access via Telnet or SSH using strong authentication techniques (such as crypto cards or one time passwords). GO-Global TN can be used to navigate the enhanced authentication techniques (by hand), and once connected to the destination host provide access to services on the remote network.
- Some sites do not provide direct network access to hosts, for example, machine A can connect to machine B, but not to machine C. Machine B, however, can connect to machine C. Using Telnet or SSH, a connection from machine A to machine B to machine C can be made to carry network data successfully.

Chapter 2: Getting started with GO-Global TN

This chapter gives a quick guide to the simplest types of interactive login session using GO-Global TN. It does not cover the use of GO-Global TN's advanced tunneling capabilities. See chapter 3 for details on GO-Global TN's tunneling.

2.1 Starting a session

When you start GO-Global TN, you will see a dialog box. This dialog box allows you to control everything GO-Global TN can do. See chapter 5 for details of all the things you can control.

You don't usually need to change most of the configuration options. To start the simplest kind of session, all you need to do is to enter a few basic parameters.

In the ‘Host Name’ box, enter the Internet host name of the server you want to connect to. You should have been told this by the provider of your login account.

Now select a login protocol to use, from the ‘Protocol’ buttons. For a login session, you should select Telnet, Rlogin or SSH. See section 1.2 for a description of the differences between the three protocols, and advice on which one to use. The fourth protocol, *Raw*, is not used for interactive login sessions; you would usually use this for debugging other Internet services.

When you change the selected protocol, the number in the ‘Port’ box will change. This is normal: it happens because the various login services are usually provided on different network ports by the server machine. Most servers will use the standard port numbers, so you will not need to change the port setting. If your server provides login services on a non-standard port, your system administrator should have told you which one. (For example, many MUDs run Telnet service on a port other than 23.)

Once you have filled in the ‘Host Name’, ‘Protocol’, and possibly ‘Port’ settings, you are ready to connect. Press the ‘Open’ button at the bottom of the dialog box, and GO-Global TN will begin trying to connect you to the server.

2.2 Verifying the Host Key (SSH only)

If you are not using the SSH protocol, you can skip this section.

If you are using SSH to connect to a server for the first time, you will see a message looking something like this:

```
The server's host key is not cached in the registry. You
have no guarantee that the server is the computer you
think it is.
The server's key fingerprint is:
ssh-rsa 1024 7b:e5:6f:a7:f4:f9:81:62:5c:e3:1f:bf:8b:57:6c:5a
If you trust this host, hit Yes to add the key to
```

```
GO-Global TN's cache and carry on connecting.
If you want to carry on connecting just once, without
adding the key to the cache, hit No.
If you do not trust this host, hit Cancel to abandon the
connection.
```

This is a feature of the SSH protocol. It is designed to protect you against a network attack known as *spoofing*: secretly redirecting your connection to a different computer, so that you send your password to the wrong machine. Using this technique, an attacker would be able to learn the password that guards your login account, and could then log in as if they were you and use the account for their own purposes.

To prevent this attack, each server has a unique identifying code, called a *host key*. These keys are created in a way that prevents one server from forging another server's key. So if you connect to a server and it sends you a different host key from the one you were expecting, GO-Global TN can warn you that the server may have been switched and that a spoofing attack might be in progress.

GO-Global TN records the host key for each server you connect to, in the Windows Registry. Every time you connect to a server, it checks that the host key presented by the server is the same host key as it was the last time you connected. If it is not, you will see a warning, and you will have the chance to abandon your connection before you type any private information (such as a password) into it.

However, when you connect to a server you have not connected to before, GO-Global TN has no way of telling whether the host key is the right one or not. So it gives the warning shown above, and asks you whether you want to trust this host key or not.

Whether or not to trust the host key is your choice. If you are connecting within a company network, you might feel that all the network users are on the same side and spoofing attacks are unlikely, so you might choose to trust the key without checking it. If you are connecting across a hostile network (such as the Internet), you should check with your system administrator, perhaps by telephone or in person. (Some modern servers have more than one host key. If the system administrator sends you more than one fingerprint, you should make sure the one GO-Global TN shows you is on the list, but it doesn't matter which one it is.)

2.3 Logging In

After you have connected, and perhaps verified the server's host key, you will be asked to log in, probably using a username and a password. Your system administrator should have provided you with these. Enter the username and the password, and the server should grant you access and begin your session. If you have mistyped your password, most servers will give you several chances to get it right.

If you are using SSH, be careful not to type your username wrongly, because you will not have a chance to correct it after you press Return. This is an unfortunate feature of the SSH protocol: it does not allow you to make two login attempts using different usernames. If you type your username wrongly, you must close GO-Global TN and start again.

If your password is refused but you are sure you have typed it correctly, check that Caps Lock is not enabled. Many login servers, particularly Unix computers, treat upper case and lower case as different when checking your password; so if Caps Lock is on, your password will probably be refused.

2.4 After Logging In

After you log in to the server, what happens next is up to the server! Most servers will print some sort of login message and then present a prompt, at which you can type commands which the server will carry out. Some servers will offer you on-line help; others might not. If you are in doubt about what to do next, consult your system administrator.

2.5 Logging Out

When you have finished your session, you should log out by typing the server's own logout command. This might vary between servers; if in doubt, try `logout` or `exit`, or consult a manual or your system administrator. When the server processes your logout command, the GO-Global TN window should close itself automatically.

You *can* close a GO-Global TN session using the Close button in the window border, but this might confuse the server - a bit like hanging up a telephone unexpectedly in the middle of a conversation. We recommend you do not do this unless the server has stopped responding to you and you cannot close the window any other way.

Chapter 3: Tunneling with GO-Global TN

This chapter describes the use of GO-Global TN's tunneling functionality.

GO-Global TN comes with a host component (called ‘ggtn’ here) that enables it to tunnel data over the login session. This is similar to SSH's tunneling abilities (see section 5.19 describing SSH tunnels and section 4.4 describing SSH's X11 tunneling), but does not require direct network connectivity between the PC and UNIX systems, and currently tunnels only from the PC to the host.

3.1 Configuring GO-Global TN Tunnels

GO-Global TN's tunnel configuration is performed on the UNIX side by editing a configuration file used by the GO-Global TN server process. The server will look in `$HOME/.ggtnrc` and in `/usr/local/etc/ggtnrc` for its configuration file, using the first one it finds. Usually, the system administrator will create a default configuration and place it into `/usr/local/etc/ggtnrc`. Individual users who wish to customize the configuration can copy the system-wide default file into their home directory and make local modifications.

3.1.1 Format of the Configuration File

The GO-Global TN configuration file format resembles that of a Windows-style `.INI` file. It contains ‘sections’ that configure tunnels, like:

```
[Tag]
ClientPort=<port#>
ServerPort=<port#>
Server=<hostname>
Launch=<path>
Probe=<true/false>
```

The `[Tag]` value is not really important and is mostly to provide a simple way to identify the tunnel being created.

The `ClientPort` parameter determines the port on which GO-Global TN should listen for incoming connections on the Windows PC. This option should not be specified if using the `Launch` parameter.

`Server` and `ServerPort` are optional, and default to the value of `ClientPort` on ‘localhost’ (localhost here referring to the UNIX system, where ggtn is running). These options determine where the UNIX ggtn process will tunnel the data to. `ServerPort` should be specified if the `Launch` parameter is specified.

The `Launch` value, if present, causes the client to create the specified tunnel, using a dynamically allocated `ClientPort` value, and then launch the command line specified in the value of the Launch parameter. This can be used to launch additional software on the PC that can take advantage of the tunnels that are created.

The Launch value can be the path to any executable on the client. To facilitate the use of this option with other GraphOn products, GO-Global TN performs the following substitutions on the command line:

- %G is replaced with the path to the GO-Global UX v2.x binary installed on the system.
- %B is replaced with the path to the GO-Global UX v1.x or Bridges for UNIX v1.x binary installed on the system.
- %p is replaced with the dynamically allocated port number, so the launched program will know where to connect to.

The Probe option will cause the ggtn server process to attempt to connect to the ServerPort before establishing the tunnel. If the connection cannot be made, the server will disable that tunnel. Probe defaults to true (even if not specified).

3.1.2 The [Globals] Section

The GO-Global TN configuration file also contains a special section that sets application preferences. This section is labeled as the ‘Global’ section. It has the following parameters:

- LogFile, which sets the path to the GO-Global TN log file. This file contains messages logged by the GO-Global TN server.
- DebugLevel, which determines which classes of messages are logged to the LogFile.
- ConsoleLevel, which determines which classes of messages are logged to the user's display.

By default, no log file is created, and only warnings, errors, and fatal errors are logged to the user's display. DebugLevel and ConsoleLevel can be set to any of the following values:

- 1, *Debug Level Critical*, log only fatal error messages.
- 2, *Debug Level Error*, log all error messages.
- 3, *Debug Level Warning*, log all warnings and errors.
- 4, *Debug Level Info*, log errors, warnings, and status messages.
- 5, *Debug Level Debug*, all of the above with limited debugging output.
- 6, *Debug Level More Debug*, everything, including copious debugging.

By default, DebugLevel is set to 4, *Debug Level Info*, and ConsoleLevel is set to 3, *Debug Level Warning*.

3.1.3 Sample Configurations

In the following configurations, the client programs on the Windows PC should be configured to connect to ‘localhost’ on the PC. When the program connects to the *ClientPort* on the PC, data sent on that connection will be forwarded through the tunnel.

To tunnel telnet from the client to the host:

```
[Telnet]
ClientPort=23
```

Then, telnet to ‘localhost’ on port 23, to connect to the telnet server on the remote UNIX host. (`Server` will default to ‘localhost’, `ServerPort` will default to port 23.)

To tunnel IMAP (Internet Mail Access Protocol) from the client to the mail server (not necessarily the machine running `ggtn`):

```
[IMAP]
ClientPort=143
Server=mailserver
```

Then, configured your IMAP client to use ‘localhost’ as its IMAP server, and the IMAP protocol will be forwarded through the tunnel to *mailserver*. (`ServerPort` will default to port 143.)

To tunnel GO-Global UX v2.x from the client to the host:

```
[GGUX]
ServerPort=491
Launch=%G host=localhost port=%p transport=libtcpip.btu
```

(`ClientPort` defaults to a dynamically allocated port, `Server` defaults to ‘localhost’, `Launch` gets filled-in with the path to GO-Global UX and the dynamically allocated port number.)

To tunnel GO-Global UX v2.x from the client to the host over SSL:

```
[GGUX-ssl]
ServerPort=791
Launch=%G host=localhost port=%p transport=libssl.btu
```

(Same as above, but this time point to the SSL port with `ServerPort` and specify different transport for GO-Global UX.)

When using GO-Global UX, the available transports are:

- `libtcpip.btu` is the standard, unencrypted TCP/IP transport.
- `libssl.btu` is the encrypted (SSL) transport.
- `libhttpsproxy.btu` is the unencrypted HTTPS-proxy tunneling transport (should probably not be used with GO-Global TN).

3.1.4 Notes on Configuring GO-Global TN

When configuring GO-Global TN tunnels, keep the following points in mind:

- `Server` and `ServerPort` specify the server and port that the data should be tunneled to. Normally, it is a port on ‘localhost’, but it can be any server that the ggtn host can connect to.
- If the client connects to its end of the tunnel, and the host is unable to complete the tunnel, it will simply close the client's connection. To the client, it will appear that it successfully connected, and it's error message(s) may reflect this.

- Make sure that when launching GO-Global UX, you specify the transport to be used. If your default transport does not match the port you are tunneling to, GO-Global will not connect successfully.

3.2 Starting GO-Global TN's Tunnels

To start tunneling, connect to the remote host and log in, then run the `ggtn` program on the remote host. The tunnels will be established and any client-software (specified in `Launch` parameters) will be started. After the tunnels have been established, the host program will launch a sub-shell for your convenience. Exiting that sub-shell will close down the GO-Global TN tunnels and exit the host module.

Chapter 4: Using GO-Global TN

This chapter provides a general introduction to some more advanced features of GO-Global TN. For extreme detail and reference purposes, chapter 5 is likely to contain more information.

4.1 During your session

A lot of GO-Global TN's complexity and features are in the configuration panel. Once you have worked your way through that and started a session, things should be reasonably simple after that. Nevertheless, there are a few more useful features available.

4.1.1 Copying and pasting text

Often in a GO-Global TN session you will find text on your terminal screen which you want to type in again. Like most other terminal emulators, GO-Global TN allows you to copy and paste the text rather than having to type it again. Also, copy and paste uses the Windows clipboard, so that you can paste (for example) URLs into a web browser, or paste from a word processor or spreadsheet into your terminal session.

GO-Global TN's copy and paste works entirely with the mouse. In order to copy text to the clipboard, you just click the left mouse button in the terminal window, and drag to select text. When you let go of the button, the text is *automatically* copied to the clipboard. You do not need to press Ctrl-C or Ctrl-Ins; in fact, if you do press Ctrl-C, GO-Global TN will send a Ctrl-C character down your session to the server where it will probably cause a process to be interrupted.

Pasting is done using the right button (or the middle mouse button, if you have a three-button mouse and have set it up; see section 5.11.2). When you click the right mouse button, GO-Global TN will read whatever is in the Windows Clipboard and paste it into your session, *exactly* as if it had been typed at the keyboard. (Therefore, be careful of pasting formatted text into an editor that does automatic indenting; you may find that the spaces pasted from the clipboard plus the spaces added by the editor add up to too many spaces and ruin the formatting. There is nothing GO-Global TN can do about this.)

If you double-click the left mouse button, GO-Global TN will select a whole word. If you double-click, hold down the second click, and drag the mouse, GO-Global TN will select a sequence of whole words. (You can adjust precisely what GO-Global TN considers to be part of a word; see section 5.11.5.) If you *triple*-click, or triple-click and drag, then GO-Global TN will select a whole line or sequence of lines.

If you want to select a rectangular region instead of selecting to the end of each line, you can do this by holding down Alt when you make your selection. (You can also configure rectangular selection to be the default, and then holding down Alt gives the normal behaviour instead. See section 5.11.4 for details.)

If you have a middle mouse button, then you can use it to adjust an existing selection if you selected something slightly wrong. (If you have configured the middle mouse button to paste, then the right mouse button does this instead.) Click the button on the screen, and you can pick up the nearest end of the selection and drag it to somewhere else.

4.1.2 Scrolling the screen back

GO-Global TN keeps track of text that has scrolled up off the top of the terminal. So if something appears on the screen that you want to read, but it scrolls too fast and it's gone by the time you try to look for it, you can use the scrollbar on the right side of the window to look back up the session history and find it again.

As well as using the scrollbar, you can also page the scrollback up and down by pressing Shift-PgUp and Shift-PgDn. These are still available if you configure the scrollbar to be invisible.

By default the last 200 lines scrolled off the top are preserved for you to look at. You can increase (or decrease) this value using the configuration box; see section 5.7.3.

4.1.3 The System menu

If you click the left mouse button on the icon in the top left corner of GO-Global TN's window, or click the right mouse button on the title bar, you will see the standard Windows system menu containing items like Minimise, Move, Size and Close.

GO-Global TN's system menu contains extra program features in addition to the Windows standard options. These extra menu commands are described below.

4.1.3.1 The GO-Global TN Event Log

If you choose ‘Event Log’ from the system menu, a small window will pop up in which GO-Global TN logs significant events during the connection. Most of the events in the log will probably take place during session startup (except for some tunneling events), but a few can occur at any point in the session, and one or two occur right at the end.

You can use the mouse to select one or more lines of the Event Log, and hit the Copy button to copy them to the clipboard. If you are reporting a bug, it's often useful to paste the contents of the Event Log into your bug report.

4.1.3.2 Starting new sessions

GO-Global TN's system menu provides some shortcut ways to start new sessions:

- Selecting ‘New Session’ will start a completely new instance of GO-Global TN, and bring up the configuration box as normal.
- Selecting ‘Duplicate Session’ will start a session with precisely the same options as your current one - connecting to the same host using the same protocol, with all the same terminal settings and everything.
- The ‘Saved Sessions’ submenu gives you quick access to any sets of stored session details you have previously saved. See section 5.1.2 for details of how to create saved sessions.

4.1.3.3 Changing your session settings

If you select ‘Change Settings’ from the system menu, GO-Global TN will display a cut-down version of its initial configuration box. This allows you to adjust most properties of your current session. You can change the terminal size, the font, the actions of various keypresses, the colours, and so on.

Some of the options that are available in the main configuration box are not shown in the cut-down Change Settings box. These are usually options which don't make sense to change in the middle of a session (for example, you can't switch from SSH to Telnet in mid-session).

4.1.3.4 Copy All to Clipboard

This system menu option provides a convenient way to copy the whole contents of the terminal screen and scrollback to the clipboard in one go.

4.1.3.5 Clearing and resetting the terminal

The ‘Clear Scrollback’ option on the system menu tells GO-Global TN to discard all the lines of text that have been kept after they scrolled off the top of the screen. This might be useful, for example, if you displayed sensitive information and wanted to make sure nobody could look over your shoulder and see it. (Note that this only prevents a casual user from using the scrollbar to view the information; the text is not guaranteed not to still be in GO-Global TN's memory.)

The ‘Reset Terminal’ option causes a full reset of the terminal emulation. A VT-series terminal is a complex piece of software and can easily get into a state where all the text printed becomes unreadable. (This can happen, for example, if you accidentally output a binary file to your terminal.) If this happens, selecting Reset Terminal should sort it out.

4.1.3.6 Full screen mode

If you find the title bar on a maximised window to be ugly or distracting, you can select Full Screen mode to maximise GO-Global TN ‘even more’. When you select this, GO-Global TN will expand to fill the whole screen and its borders, title bar and scrollbar will disappear. (You can configure the scrollbar not to disappear in full-screen mode if you want to keep it; see section 5.7.3.)

When you are in full-screen mode, you can still access the system menu if you click the left mouse button in the *extreme* top left corner of the screen.

4.2 Creating a log file of your session

For some purposes you may find you want to log everything that appears on your screen. You can do this using the ‘Logging’ panel in the configuration box.

To begin a session log, select ‘Change Settings’ from the system menu and go to the Logging panel. Enter a log file name, and select a logging mode. (You can log all session output including the terminal control sequences, or you can just log the printable text. It depends what you want the log for.) Click ‘Apply’ and your log will be started. Later on, you can go back to the Logging panel and select ‘Logging turned off completely’ to stop logging; then GO-Global TN will close the log file and you can safely read it.

See section 5.2 for more details and options.

4.3 Altering your character set configuration

If you find that special characters (accented characters, for example) are not being displayed correctly in your GO-Global TN session, it may be that GO-Global TN is interpreting the characters sent by the server according to the wrong *character set*. There are a lot of different character sets available, so it's entirely possible for this to happen.

If you click ‘Change Settings’ and look at the ‘Translation’ panel, you should see a large number of character sets which you can select. Now all you need is to find out which of them you want!

4.4 Using X11 forwarding in SSH

The SSH protocol has the ability to securely forward X Window System applications over your encrypted SSH connection, so that you can run an application on the SSH server machine and have it put its windows up on your local machine without sending any X network traffic in the clear.

In order to use this feature, you will need an X display server for your Windows machine. This will probably install itself as display number 0 on your local machine; if it doesn't, the manual for the X server should tell you what it does do.

You should then tick the ‘Enable X11 forwarding’ box in the Tunnels panel (see section 5.19.1) before starting your SSH session. The ‘X display location’ box reads `localhost:0` by default, which is the usual display location where your X server will be installed. If that needs changing, then change it.

Now you should be able to log in to the SSH server as normal. To check that X forwarding has been successfully negotiated during connection startup, you can check the GO-Global TN Event Log (see section 4.1.3.1). It should say something like this:

```
2001-12-05 17:22:01 Requesting X11 forwarding
2001-12-05 17:22:02 X11 forwarding enabled
```

If the remote system is Unix or Unix-like, you should also be able to see that the `DISPLAY` environment variable has been set to point at display 10 or above on the SSH server machine itself:

```
fred@unixbox:~$ echo $DISPLAY
unixbox:10.0
```

If this works, you should then be able to run X applications in the remote session and have them display their windows on your PC.

Note that if your PC X server requires authentication to connect, then GO-Global TN cannot currently support it. If this is a problem for you, you should mail the authors and give details.

4.5 Using port forwarding in SSH

The SSH protocol has the ability to forward arbitrary network connections over your encrypted SSH connection, to avoid the network traffic being sent in clear. For example, you could use this to connect from your home computer to a POP-3 server on a remote machine without your POP-3 password being visible to network sniffers.

In order to use port forwarding to connect from your local machine to a port on a remote server, you need to:

- Choose a port number on your local machine where GO-Global TN should listen for incoming connections. There are likely to be plenty of unused port numbers above 3000.
- Now, before you start your SSH connection, go to the Tunnels panel (see section 5.19.2). Make sure the ‘Local’ radio button is set. Enter the local port number into the ‘Source port’ box. Enter the destination host name and port number into the ‘Destination’ box, separated by a colon (for example, `popserver.example.com:110` to connect to a POP-3 server).
- Now click the ‘Add’ button. The details of your port forwarding should appear in the list box.

Now start your session and log in. (Port forwarding will not be enabled until after you have logged in; otherwise it would be easy to perform completely anonymous network attacks, and gain access to anyone's virtual private network). To check that GO-Global TN has set up the port forwarding correctly, you can look at the GO-Global TN Event Log (see section 4.1.3.1). It should say something like this:

```
2001-12-05 17:22:10 Local port 3110 forwarding to
         popserver.example.com:110
```

Now if you connect to the source port number on your local PC, you should find that it answers you exactly as if it were the service running on the destination machine. So in this example, you could then configure an e-mail client to use `localhost:3110` as a POP-3 server instead of `popserver.example.com:110`. (Of course, the forwarding will stop happening when your GO-Global TN session closes down.)

You can also forward ports in the other direction: arrange for a particular port number on the *server* machine to be forwarded back to your PC as a connection to a service on your PC or near it. To do this, just select the ‘Remote’ radio button instead of the ‘Local’ one. The ‘Source port’ box will now specify a port number on the *server* (note that most servers will not allow you to use port numbers under 1024 for this purpose).

The source port for a forwarded connection usually does not accept connections from any machine except the SSH client or server machine itself (for local and remote forwardings respectively). There are controls in the Tunnels panel to change this:

- The ‘Local ports accept connections from other hosts’ option allows you to set up local-to-remote port forwardings in such a way that machines other than your client PC can connect to the forwarded port.
- The ‘Remote ports do the same’ option does the same thing for remote-to-local port forwardings (so that machines other than the SSH server machine can connect to the forwarded port.) Note that this feature is only available in the SSH 2 protocol, and not all SSH 2 servers support it (OpenSSH 3.0 does not, for example).

4.6 Making raw TCP connections

A lot of Internet protocols are composed of commands and responses in plain text. For example, SMTP (the protocol used to transfer e-mail), NNTP (the protocol used to transfer Usenet news), and HTTP (the protocol used to serve Web pages) all consist of commands in readable plain text.

Sometimes it can be useful to connect directly to one of these services and speak the protocol ‘by hand’, by typing protocol commands and watching the responses. On Unix machines, you can do this using the system's `telnet` command to connect to the right port number. For example, `telnet mailserver.example.com 25` might enable you to talk directly to the SMTP service running on a mail server.

Although the Unix `telnet` program provides this functionality, the protocol being used is not really Telnet. Really there is no actual protocol at all; the bytes sent down the connection are exactly the ones you type, and the bytes shown on the screen are exactly the ones sent by the server. Unix `telnet` will attempt to detect or guess whether the service it is talking to is a real Telnet service or not; GO-Global TN prefers to be told for certain.

In order to make a debugging connection to a service of this type, you simply select the fourth protocol name, ‘Raw’, from the ‘Protocol’ buttons in the ‘Session’ configuration panel. (See section 5.1.1.) You can then enter a host name and a port number, and make the connection.

Chapter 5: Configuring GO-Global TN

This chapter describes all the configuration options in GO-Global TN.

GO-Global TN is configured using the control panel that comes up before you start a session. Some options can also be changed in the middle of a session, by selecting ‘Change Settings’ from the window menu.

5.1 The Session panel

The Session configuration panel contains the basic options you need to specify in order to open a session at all, and also allows you to save your settings to be reloaded later.

5.1.1 The host name section

The top box on the Session panel, labelled ‘Specify your connection by host name’, contains the details that need to be filled in before GO-Global TN can open a session at all.

- The ‘Host Name’ box is where you type the name, or the IP address, of the server you want to connect to.
- The ‘Protocol’ radio buttons let you choose what type of connection you want to make: a raw connection, a Telnet connection, an rlogin connection or an SSH connection. (See section 1.2 for a summary of the differences between SSH, Telnet and rlogin.)
- The ‘Port’ box lets you specify which port number on the server to connect to. If you select Telnet, Rlogin, or SSH, this box will be filled in automatically to the usual value, and you will only need to change it if you have an unusual server. If you select Raw mode (see section 4.6), you will almost certainly need to fill in the ‘Port’ box.

5.1.2 Loading and storing saved sessions

The next part of the Session configuration panel allows you to save your preferred GO-Global TN options so they will appear automatically the next time you start GO-Global TN. It also allows you to create *saved sessions*, which contain a full set of configuration options plus a host name and protocol. A saved session contains all the information GO-Global TN needs to start exactly the session you want.

- To save your default settings: first set up the settings the way you want them saved. Then come back to the Session panel. Select the ‘Default Settings’ entry in the saved sessions list, with a single click. Then press the ‘Save’ button.

Note that GO-Global TN does not allow you to save a host name into the Default Settings entry.

This ensures that when GO-Global TN is started up, the host name box is always empty, so a user can always just type in a host name and connect.

If there is a specific host you want to store the details of how to connect to, you should create a saved session, which will be separate from the Default Settings.

- To save a session: first go through the rest of the configuration box setting up all the options you want. Then come back to the Session panel. Enter a name for the saved session in the 'Saved Sessions' input box. (The server name is often a good choice for a saved session name.) Then press the 'Save' button. Your saved session name should now appear in the list box.

- To reload a saved session: single-click to select the session name in the list box, and then press the 'Load' button. Your saved settings should all appear in the configuration panel.

- To modify a saved session: first load it as described above. Then make the changes you want. Come back to the Session panel, single-click to select the session name in the list box, and press the 'Save' button. The new settings will be saved over the top of the old ones.

- To start a saved session immediately: double-click on the session name in the list box.

- To delete a saved session: single-click to select the session name in the list box, and then press the 'Delete' button.

Each saved session is independent of the Default Settings configuration. If you change your preferences and update Default Settings, you must also update every saved session separately.

Saved sessions are stored in the Registry, at the location

```
HKEY_CURRENT_USER\Software\Graphon\GO-Global TN\Sessions
```

If you need to store them in a file, you could try the method described in section 5.21.

5.1.3 'Close Window on Exit'

Finally in the Session panel, there is an option labelled 'Close Window on Exit'. This controls whether the GO-Global TN session window disappears as soon as the session inside it terminates. If you are likely to want to copy and paste text out of the session after it has terminated, you should arrange this option to be off.

'Close Window On Exit' has three settings. 'Always' means always close the window on exit; 'Never' means never close on exit (always leave the window open). The third setting, and the default one, is 'Only on clean exit'. In this mode, a session which terminates normally will cause its window to close, but one which is aborted unexpectedly by network trouble or a confusing message from the server will leave the window up.

5.2 The Logging panel

The Logging configuration panel allows you to save log files of your GO-Global TN sessions, for debugging, analysis or future reference.

The main option is a radio-button set that specifies whether GO-Global TN will log anything at all. The options are

- ‘Logging turned off completely’. This is the default option; in this mode GO-Global TN will not create a log file at all.
- ‘Log printable output only’. In this mode, a log file will be created and written to, but only printable text will be saved into it. The various terminal control codes that are typically sent down an interactive session alongside the printable text will be omitted. This might be a useful mode if you want to read a log file in a text editor and hope to be able to make sense of it.
- ‘Log all session output’. In this mode, *everything* sent by the server into your terminal session is logged. If you view the log file in a text editor, therefore, you may well find it full of strange control characters. This is a particularly useful mode if you are experiencing problems with GO-Global TN's terminal handling: you can record everything that went to the terminal, so that someone else can replay the session later in slow motion and watch to see what went wrong.
- ‘Log SSH packet data’. In this mode (which is only used by SSH connections), the SSH message packets sent over the encrypted connection are written to the log file. You might need this to debug a network-level problem, or more likely to send to the GO-Global TN authors as part of a bug report. *BE WARNED* that if you log in using a password, the password will appear in the log file, so be sure to edit it out before sending the log file to anyone else!

5.2.1 ‘Log file name’

In this edit box you enter the name of the file you want to log the session to. The ‘Browse’ button will let you look around your file system to find the right place to put the file; or if you already know exactly where you want it to go, you can just type a pathname into the edit box.

There are a few special features in this box. If you use the & character in the file name box, GO-Global TN will insert details of the current session in the name of the file it actually opens. The precise replacements it will do are:

- `&Y` will be replaced by the current year, as four digits.
- `&M` will be replaced by the current month, as two digits.
- `&D` will be replaced by the current day of the month, as two digits.
- `&T` will be replaced by the current time, as six digits (HHMMSS) with no punctuation.
- `&H` will be replaced by the host name you are connecting to.

For example, if you enter the host name `c:\ggtnlogs\log-&h-&y&m&d-&t.dat`, you will end up with files looking like

```
log-server1.example.com-20010528-110859.dat
log-unixbox.somewhere.org-20010611-221001.dat
```

5.2.2 ‘What to do if the log file already exists’

This control allows you to specify what GO-Global TN should do if it tries to start writing to a log file and it finds the file already exists. You might want to automatically destroy the existing log file and start a new one with the same name. Alternatively, you might want to open the existing log file and add data to the *end* of it. Finally (the default option), you might not want to have any automatic behaviour, but to ask the user every time the problem comes up.

5.3 The Terminal panel

The Terminal configuration panel allows you to control the behaviour of GO-Global TN's terminal emulation.

5.3.1 ‘Auto wrap mode initially on’

Auto wrap mode controls what happens when text printed in a GO-Global TN window reaches the right-hand edge of the window.

With auto wrap mode on, if a long line of text reaches the right-hand edge, it will wrap over on to the next line so you can still see all the text. With auto wrap mode off, the cursor will stay at the right-hand edge of the screen, and all the characters in the line will be printed on top of each other.

If you are running a full-screen application and you occasionally find the screen scrolling up when it looks as if it shouldn't, you could try turning this option off.

Auto wrap mode can be turned on and off by control sequences sent by the server. This configuration option only controls the *default* state, which will be restored when you reset the terminal (see section 4.1.3.5). However, if you modify this option in mid-session using ‘Change Settings’, it will take effect immediately.

5.3.2 ‘DEC Origin Mode initially on’

DEC Origin Mode is a minor option which controls how GO-Global TN interprets cursor-position control sequences sent by the server.

The server can send a control sequence that restricts the scrolling region of the display. For example, in an editor, the server might reserve a line at the top of the screen and a line at the bottom, and might send a control sequence that causes scrolling operations to affect only the remaining lines.

With DEC Origin Mode on, cursor coordinates are counted from the top of the scrolling region. With it turned off, cursor coordinates are counted from the top of the whole screen regardless of the scrolling region.

It is unlikely you would need to change this option, but if you find a full-screen application is displaying pieces of text in what looks like the wrong part of the screen, you could try turning DEC Origin Mode on to see whether that helps.

DEC Origin Mode can be turned on and off by control sequences sent by the server. This configuration option only controls the *default* state, which will be restored when you reset the terminal (see section 4.1.3.5). However, if you modify this option in mid-session using ‘Change Settings’, it will take effect immediately.

5.3.3 ‘Implicit CR in every LF’

Most servers send two control characters, CR and LF, to start a new line of the screen. The CR character makes the cursor return to the left-hand side of the screen. The LF character makes the cursor move one line down (and might make the screen scroll).

Some servers only send LF, and expect the terminal to move the cursor over to the left automatically. If you come across a server that does this, you will see a stepped effect on the screen, like this:

```
First line of text
                  Second line
                             Third line
```

If this happens to you, try enabling the ‘Implicit CR in every LF’ option, and things might go back to normal:

```
First line of text
Second line
Third line
```

5.3.4 ‘Use background colour to erase screen’

Not all terminals agree on what colour to turn the screen when the server sends a ‘clear screen’ sequence. Some terminals believe the screen should always be cleared to the *default* background colour. Others believe the screen should be cleared to whatever the server has selected as a background colour.

There exist applications that expect both kinds of behaviour. Therefore, GO-Global TN can be configured to do either.

With this option disabled, screen clearing is always done in the default background colour. With this option enabled, it is done in the *current* background colour.

Background-colour erase can be turned on and off by control sequences sent by the server. This configuration option only controls the *default* state, which will be restored when you reset the terminal (see section 4.1.3.5). However, if you modify this option in mid-session using ‘Change Settings’, it will take effect immediately.

5.3.5 ‘Enable blinking text’

The server can ask GO-Global TN to display text that blinks on and off. This is very distracting, so GO-Global TN allows you to turn blinking text off completely.

When blinking text is disabled and the server attempts to make some text blink, GO-Global TN will instead display the text with a bolded background colour.

Blinking text can be turned on and off by control sequences sent by the server. This configuration option only controls the *default* state, which will be restored when you reset the terminal (see section 4.1.3.5). However, if you modify this option in mid-session using ‘Change Settings’, the changes will take effect immediately.

5.3.6 ‘Answerback to ^E’

This option controls what GO-Global TN will send back to the server if the server sends it the ^E enquiry character. Normally it just sends the string ‘GO-Global TN’.

If you accidentally write the contents of a binary file to your terminal, you will probably find that it contains more than one ^E character, and as a result your next command line will probably read ‘GO-Global TNGO-Global TNGO-Global TN...’ as if you had typed the answerback string multiple times at the keyboard. If you set the answerback string to be empty, this problem should go away, but doing so might cause other problems.

Note that this is *not* the feature of GO-Global TN which the server will typically use to determine your terminal type. That feature is the ‘Terminal-type string’ in the Connection panel; see section 5.13.1 for details.

You can include control characters in the answerback string using `^C` notation. (Use `^~` to get a literal `^`.)

5.3.7 ‘Local echo’

With local echo disabled, characters you type into the GO-Global TN window are not echoed in the window *by GO-Global TN*. They are simply sent to the server. (The *server* might choose to echo them back to you; this can't be controlled from the GO-Global TN control panel.)

Some types of session need local echo, and many do not. In its default mode, GO-Global TN will automatically attempt to deduce whether or not local echo is appropriate for the session you are working in. If you find it has made the wrong decision, you can use this configuration option to override its choice: you can force local echo to be turned on, or force it to be turned off, instead of relying on the automatic detection.

5.3.8 ‘Local line editing’

Normally, every character you type into the GO-Global TN window is sent immediately to the server the moment you type it.

If you enable local line editing, this changes. GO-Global TN will let you edit a whole line at a time locally, and the line will only be sent to the server when you press Return. If you make a mistake, you can use the Backspace key to correct it before you press Return, and the server will never see the mistake.

Since it is hard to edit a line locally without being able to see it, local line editing is mostly used in conjunction with local echo (section 5.3.7). This makes it ideal for use in raw mode or when connecting to MUDs or talkers. (Although some more advanced MUDs do occasionally turn local line editing on and turn local echo off, in order to accept a password from the user.)

Some types of session need local line editing, and many do not. In its default mode, GO-Global TN will automatically attempt to deduce whether or not local line editing is appropriate for the session you are working in. If you find it has made the wrong decision, you can use this configuration option to override its choice: you can force local line editing to be turned on, or force it to be turned off, instead of relying on the automatic detection.

5.3.9 Remote-controlled printing

A lot of VT100-compatible terminals upport printing under control of the remote server. GO-Global TN supports this feature as well, but it is turned off by default.

To enable remote-controlled printing, choose a printer from the ‘Printer to send ANSI printer output to’ drop-down list box. This should allow you to select from all the printers you have installed drivers for on your computer. Alternatively, you can type the network name of a networked printer (for example, `\\printserver\printer1`) even if you haven't already installed a driver for it on your own machine.

When the remote server attempts to print some data, GO-Global TN will send that data to the printer *raw* - without translating it, attempting to format it, or doing anything else to it. It is up to you to ensure your remote server knows what type of printer it is talking to.

Since GO-Global TN sends data to the printer raw, it cannot offer options such as portrait versus landscape, print quality, or paper tray selection. All these things would be done by your PC printer driver (which GO-Global TN bypasses); if you need them done, you will have to find a way to configure your remote server to do them.

To disable remote printing again, choose ‘None (printing disabled)’ from the printer selection list. This is the default state.

5.4 The Keyboard panel

The Keyboard configuration panel allows you to control the behaviour of the keyboard in GO-Global TN.

5.4.1 Changing the action of the Backspace key

Some terminals believe that the Backspace key should send the same thing to the server as Control-H (ASCII code 8). Other terminals believe that the Backspace key should send ASCII code 127 (usually known as Control-?) so that it can be distinguished from Control-H. This option allows you to choose which code GO-Global TN generates when you press Backspace.

If you are connecting to a Unix system, you will probably find that the Unix `stty` command lets you configure which the server expects to see, so you might not need to change which one GO-Global TN generates. On other systems, the server's expectation might be fixed and you might have no choice but to configure GO-Global TN.

If you do have the choice, we recommend configuring GO-Global TN to generate Control-? and configuring the server to expect it, because that allows applications such as `emacs` to use Control-H for help.

5.4.2 Changing the action of the Home and End keys

The Unix terminal emulator `rxvt` disagrees with the rest of the world about what character sequences should be sent to the server by the Home and End keys.

`xterm`, and other terminals, send `ESC [1~` for the Home key, and `ESC [4~` for the End key. `rxvt` sends `ESC [H` for the Home key and `ESC [Ow` for the End key.

If you find an application on which the Home and End keys aren't working, you could try switching this option to see if it helps.

5.4.3 Changing the action of the function keys and keypad

This option affects the function keys (F1 to F12) and the top row of the numeric keypad.

- In the default mode, labelled `ESC [n~`, the function keys generate sequences like `ESC [11~`, `ESC [12~` and so on. This matches the general behaviour of Digital's terminals.
- In Linux mode, F6 to F12 behave just like the default mode, but F1 to F5 generate `ESC [[A` through to `ESC [[E`. This mimics the Linux virtual console.
- In Xterm R6 mode, F5 to F12 behave like the default mode, but F1 to F4 generate `ESC OP` through to `ESC OS`, which are the sequences produced by the top row of the *keypad* on Digital's terminals.
- In VT400 mode, all the function keys behave like the default mode, but the actual top row of the numeric keypad generates `ESC OP` through to `ESC OS`.
- In VT100+ mode, the function keys generate `ESC OP` through to `ESC O[`
- In SCO mode, the function keys F1 to F12 generate `ESC [M` through to `ESC [X`. Together with shift, they generate `ESC [Y` through to `ESC [j`. With control they generate `ESC [k` through to `ESC [v`, and with shift and control together they generate `ESC [w` through to `ESC [{`.

If you don't know what any of this means, you probably don't need to fiddle with it.

5.4.4 Controlling Application Cursor Keys mode

Application Cursor Keys mode is a way for the server to change the control sequences sent by the arrow keys. In normal mode, the arrow keys send `ESC [A` through to `ESC [D`. In application mode, they send `ESC OA` through to `ESC OD`.

Application Cursor Keys mode can be turned on and off by the server, depending on the application. GO-Global TN allows you to configure the initial state.

You can also disable application cursor keys mode completely, using the 'Features' configuration panel; see section 5.6.1.

5.4.5 Controlling Application Keypad mode

Application Keypad mode is a way for the server to change the behaviour of the numeric keypad.

In normal mode, the keypad behaves like a normal Windows keypad: with NumLock on, the number keys generate numbers, and with NumLock off they act like the arrow keys and Home, End etc.

In application mode, all the keypad keys send special control sequences, *including* Num Lock. Num Lock stops behaving like Num Lock and becomes another function key.

Depending on which version of Windows you run, you may find the Num Lock light still flashes on and off every time you press Num Lock, even when application mode is active and Num Lock is acting like a function key. This is unavoidable.

Application keypad mode can be turned on and off by the server, depending on the application. GO-Global TN allows you to configure the initial state.

You can also disable application keypad mode completely, using the ‘Features’ configuration panel; see section 5.6.1.

5.4.6 Using NetHack keypad mode

GO-Global TN has a special mode for playing NetHack. You can enable it by selecting ‘NetHack’ in the ‘Initial state of numeric keypad’ control.

In this mode, the numeric keypad keys 1-9 generate the NetHack movement commands (`hjklyubn`). The 5 key generates the `.` command (do nothing).

Better still, pressing Shift with the keypad keys generates the capital forms of the commands (`HJKLYUBN`), which tells NetHack to keep moving you in the same direction until you encounter something interesting.

For some reason, this feature only works properly when Num Lock is on. We don't know why.

5.4.7 Enabling a DEC-like Compose key

DEC terminals have a Compose key, which provides an easy-to-remember way of typing accented characters. You press Compose and then type two more characters. The two characters are ‘combined’ to produce an accented character. The choices of character are designed to be easy to remember; for example, composing ‘e’ and ‘`’ produces the ‘è’ character.

If your keyboard has a Windows Application key, it acts as a Compose key in GO-Global TN. Alternatively, if you enable the ‘AltGr acts as Compose key’ option, the AltGr key will become a Compose key.

5.4.8 ‘Control-Alt is different from AltGr’

Some old keyboards do not have an AltGr key, which can make it difficult to type some characters. GO-Global TN can be configured to treat the key combination Ctrl + Left Alt the same way as the AltGr key.

By default, this checkbox is checked, and the key combination Ctrl + Left Alt does something completely different. GO-Global TN's usual handling of the left Alt key is to prefix the Escape (Control-`[`) character to whatever character sequence the rest of the keypress would generate. For example, Alt-A generates Escape followed by `a`. So Alt-Ctrl-A would generate Escape, followed by Control-A.

If you uncheck this box, Ctrl-Alt will become a synonym for AltGr, so you can use it to type extra graphic characters if your keyboard has any.

(However, Ctrl-Alt will never act as a Compose key, regardless of the setting of ‘AltGr acts as Compose key’ described in section 5.4.7.)

5.5 The Bell panel

The Bell panel controls the terminal bell feature: the server's ability to cause GO-Global TN to beep at you.

In the default configuration, when the server sends the character with ASCII code 7 (Control-G), GO-Global TN will play the Windows Default Beep sound. This is not always what you want the terminal bell feature to do; the Bell panel allows you to configure alternative actions.

5.5.1 ‘Set the style of bell’

This control allows you to select various different actions to occur on a terminal bell:

- Selecting ‘None’ disables the bell completely. In this mode, the server can send as many Control-G characters as it likes and nothing at all will happen.
- ‘Make default system alert sound’ is the default setting. It causes the Windows ‘Default Beep’ sound to be played. To change what this sound is, or to test it if nothing seems to be happening, use the Sound configurer in the Windows Control Panel.
- ‘Visual bell’ is a silent alternative to a beeping computer. In this mode, when the server sends a Control-G, the whole GO-Global TN window will flash white for a fraction of a second.
- ‘Beep using the PC speaker’ is self-explanatory.
- ‘Play a custom sound file’ allows you to specify a particular sound file to be used by GO-Global TN alone, or even by a particular individual GO-Global TN session. This allows you to distinguish your GO-Global TN beeps from any other beeps on the system. If you select this option, you will also need to enter the name of your sound file in the edit control ‘Custom sound file to play as a bell’.

5.5.2 ‘Taskbar/caption indication on bell’

This feature controls what happens to the GO-Global TN window's entry in the Windows Taskbar if a bell occurs while the window does not have the input focus.

In the default state (‘Disabled’) nothing unusual happens.

If you select ‘Steady’, then when a bell occurs and the window is not in focus, the window's Taskbar entry and its title bar will change colour to let you know that GO-Global TN session is asking for your attention. The change of colour will persist until you select the window, so you can leave several GO-Global TN windows minimised in your terminal, go away from your keyboard, and be sure not to have missed any important beeps when you get back.

‘Flashing’ is even more eye-catching: the Taskbar entry will continuously flash on and off until you select the window.

5.5.3 ‘Control the bell overload behaviour’

A common user error in a terminal session is to accidentally run the Unix command `cat` (or equivalent) on an inappropriate file type, such as an executable, image file, or ZIP file. This produces a huge stream of non-text characters sent to the terminal, which typically includes a lot of bell characters. As a result of this the terminal often doesn't stop beeping for ten minutes, and everybody else in the office gets annoyed.

To try to avoid this behaviour, or any other cause of excessive beeping, GO-Global TN includes a bell overload management feature. In the default configuration, receiving more than five bell characters in a two-second period will cause the overload feature to activate. Once the overload feature is active, further bells will have no effect at all, so the rest of your binary file will be sent to the screen in silence. After a period of five seconds during which no further bells are received, the overload feature will turn itself off again and bells will be re-enabled.

If you want this feature completely disabled, you can turn it off using the checkbox ‘Bell is temporarily disabled when over-used’.

Alternatively, if you like the bell overload feature but don't agree with the settings, you can configure the details: how many bells constitute an overload, how short a time period they have to arrive in to do so, and how much silent time is required before the overload feature will deactivate itself.

Bell overload mode is always deactivated by any keypress in the terminal. This means it can respond to large unexpected streams of data, but does not interfere with ordinary command-line activities that generate beeps (such as filename completion).

5.6 The Features panel

GO-Global TN's terminal emulation is very highly featured, and can do a lot of things under remote server control. Some of these features can cause problems due to buggy or strangely configured server applications.

The Features configuration panel allows you to disable some of GO-Global TN's more advanced terminal features, in case they cause trouble.

5.6.1 Disabling application keypad and cursor keys

Application keypad mode (see section 5.4.5) and application cursor keys mode (see section 5.4.4) alter the behaviour of the keypad and cursor keys. Some applications enable these modes but then do not deal correctly with the modified keys. You can force these modes to be permanently disabled no matter what the server tries to do.

5.6.2 Disabling `xterm`-style mouse reporting

GO-Global TN allows the server to send control codes that let it take over the mouse and use it for purposes other than copy and paste. Applications which use this feature niclude the text-mode web browser `links`, the Usenet newsreader `trn` version 4, and the file manager `mc` (Midnight Commander).

If you find this feature inconvenient, you can disable it using the ‘Disable xterm-style mouse reporting’ control. With this box ticked, the mouse will *always* do copy and paste in the normal way.

Note that even if the application takes over the mouse, you can still manage GO-Global TN's copy and paste by holding down the Shift key while you select and paste, unless you have deliverately turned this feature off (see section 5.11.3).

5.6.3 Disabling remote terminal resizing

GO-Global TN has the ability to change the terminal's size and position in response to commands from the server. If you find GO-Global TN is doing this unexpectedly or inconveniently, you can tell GO-Global TN not to respond to those server commands.

5.6.4 Disabling switching to the alternate screen

Many terminals, including GO-Global TN, support an ‘alternate screen’. This is the same size as the ordinary terminal screen, but separate. Typically a screen-based program such as a text editor might switch the terminal to the alternate screen before starting up. Then at the end of the run, it switches back to the primary screen, and you see the screen contents just as they were before starting the editor.

Some people prefer this not to happen. If you want your editor to run in the same screen as the rest of your terminal activity, you can disable the alternate screen feature completely.

5.6.5 Disabling remote window title changing

GO-Global TN has the ability to change the window title in response to commands from the server. If you find GO-Global TN is doing this unexpectedly or inconveniently, you can tell GO-Global TN not to respond to those server commands.

5.6.6 Disabling remote window title querying

GO-Global TN can optionally provide the xterm service of allowing server applications to find out the local window title. This feature is disabled by default, but you can turn it on if you really want it.

NOTE that this feature is a *potential security hazard*. If a malicious application can wrhite data to your terminal (for example, if you merely `cat` a file owned by someone else on the server machine), it can change your window title (unless you have disabled this as mentioned in section 5.6.5) and then use this service to have the new window title sent back to the server as if typed at the keyboard. This allows an attacker to fake keypresses and potentially cause your server-side applications to do things you didn't want. therefore, this feature is disabled by default, and we recommend you do not turn it on unless you *really* know what you are doing.

5.6.7 Disabling destructive backspace

Normally, when GO-Global TN receives character 127 (^?) from the server, it will perform a ‘destructive backspace’: move the cursor one space left and delete the character under it. This can apparently cause problems in some applications, so GO-Global TN provides the ability to configure character 127 to perform a normal backspace (without deleting a character) instead.

5.6.8 Disabling remote character set configuration

GO-Global TN has the ability to change its character set configuration in response to commands from the server. Some programs send these commands unexpectedly or inconveniently. In particular, BitchX (an IRC client) seems to have a habit of reconfiguring the character set to something other than the user intended.

If you find that accented characters are not showing up the way you expect them to, particularly if you're running BitchX, you could try disabling the remote character set configuration commands.

5.7 The Window panel

The Window configuration panel allows you to control aspects of the GO-Global TN window.

5.7.1 Setting the size of the GO-Global TN window

The ‘Rows’ and ‘Columns’ boxes let you set the GO-Global TN window to a precise size. Of course you can also drag the window to a new size while a session is running.

5.7.2 What to do when the window is resized

These options allow you to control what happens when the user tries to resize the GO-Global TN window.

When you resize the GO-Global TN window, one of four things can happen:

- Nothing (if you have completely disabled resizes).
- The font size can stay the same and the number of rows and columns in the terminal can change.
- The number of rows and columns in the terminal can stay the same, and the font size can change.
- You can allow GO-Global TN to change *either* the terminal size or the font size. In this mode it will change the terminal size most of the time, but enlarge the font when you maximise the window.

You can control which of these happens using the ‘Lock terminal size against resizing’ and ‘Lock font size against resizing’ options. If you lock both, the window will refuse to be resized at all. If you lock just the terminal size, the font size will change when you resize the window. If you lock just the font size, the terminal size will change when you resize the window.

5.7.3 Controlling scrollback

These options let you configure the way GO-Global TN keeps text after it scrolls off the top of the screen (see section 4.1.2).

The ‘Lines of scrollback’ box lets you configure how many lines of text GO-Global TN keeps. The ‘Display scrollbar’ options allow you to hide the scrollbar (although you can still view the scrollback using the keyboard as described in section 4.1.2). You can separately configure whether the scrollbar is shown in full-screen mode and in normal modes.

If you are viewing part of the scrollback when the server sends more text to GO-Global TN, the screen will revert to showing the current terminal contents. You can disable this behaviour by turning off ‘Reset scrollback on display activity’. You can also make the screen revert when you press a key, by turning on ‘Reset scrollback on keypress’.

5.7.4 ‘Push erased text into scrollback’

When this option is enabled, the contents of the terminal screen will be pushed into the scrollback when a server-side application clears the screen, so that your scrollback will contant a better record of what was on your screen in the past.

If the application switches to the alternate screen (see section 5.6.4 for more about this), then the contents of the primary screen will be visible in the scrollback until the application switches back again.

This option is enabled by default.

5.8 The Appearance panel

The Appearance configuration panel allows you to control aspects of the appearance of GO-Global TN's window.

5.8.1 Controlling the appearance of the cursor

The ‘Cursor appearance’ option lets you configure the cursor to be a block, an underline, or a vertical line. A block cursor becomes an empty box when the window loses focus; an underline or a vertical line becomes dotted.

The ‘Cursor blinks’ option makes the cursor blink on and off. This works in any of the cursor modes.

5.8.2 Controlling the font used in the terminal window

This option allows you to choose what font, in what size, the GO-Global TN terminal window uses to display the text in the session. You will be offered a choice from all the fixed-width fonts installed on the system. (VT100-style terminal handling can only deal with fixed- width fonts.)

5.8.3 ‘Hide mouse pointer when typing in window’

If you enable this option, the mouse pointer will disappear if the GO-Global TN window is selected and you press a key. This way, it will not obscure any of the text in the window while you work in your session. As soon as you move the mouse, the pointer will reappear.

This option is disabled by default, so the mouse pointer remains visible at all times.

5.8.4 Controlling the window border

GO-Global TN allows you to configure the appearance of the window border to some extent.

The checkbox marked ‘Sunken-edge border’ changes the appearance of the window border to something more like a DOS box: the inside edge of the border is highlighted as if it sank down to meet the surface inside the window. This makes the border a little bit thicker as well. It's hard to describe well. Try it and see if you like it.

You can also configure a completely blank gap between the text in the window and the border, using the ‘Gap between text and window edge’ control. By default this is set at one pixel. You can reduce it to zero, or increase it further.

5.9 The Behaviour panel

The Behaviour configuration panel allows you to control aspects of the behaviour of GO-Global TN's window.

5.9.1 Controlling the window title

The ‘Window title’ edit box allows you to set the title of the GO-Global TN window. By default the window title will contain the host name followed by ‘GO-Global TN’, for example `server1.example.com - GO-Global TN`. If you want a different window title, this is where to set it.

GO-Global TN allows the server to send `xterm` control sequences which modify the title of the window in mid-session (unless this is disabled - see section 5.6.5); the title string set here is therefore only the *initial* window title.

As well as the *window* title, there is also an `xterm` sequence to modify the title of the window's *icon*. This makes sense in a windowing system where the window becomes an icon when minimised, such as Windows 3.1 or most X Window System setups; but in the Windows 95-like user interface it isn't as applicable.

By default GO-Global TN only uses the server-supplied *window* title, and ignores the icon title entirely. If for some reason you want to see both titles, check the box marked ‘Separate window and icon titles’. If you do this, GO-Global TN's window title and Taskbar caption will change into the server-supplied icon title if you minimise the GO-Global TN window, and change back to the server-supplied window title if you restore it. (If the server has not bothered to supply a window or icon title, none of this will happen.)

5.9.2 ‘Warn before closing window’

If you press the Close button in a GO-Global TN window that contains a running session, GO-Global TN will put up a warning window asking if you really meant to close the window. A window whose session has already terminated can always be closed without a warning.

If you want to be able to close a window quickly, you can disable the ‘Warn before closing window’ option.

5.9.3 ‘Window closes on ALT-F4’

By default, pressing ALT-F4 causes the window to close (or a warning box to appear; see section 5.9.2). If you disable the ‘Window closes on ALT-F4’ option, then pressing ALT-F4 will simply send a key sequence to the server.

5.9.4 ‘System menu appears on ALT-Space’

If this option is enabled, then pressing ALT-Space will bring up the GO-Global TN window's menu, like clicking on the top left corner. If it is disabled, then pressing ALT-Space will just send `ESC SPACE` to the server.

Some accessibility programs for Windows may need this option enabling to be able to control GO-Global TN's window successfully. For instance, Dragon NaturallySpeaking requires it both to open the system menu via voice, and to close, minimise, maximise and restore the window.

5.9.5 ‘System menu appears on Alt alone’

If this option is enabled, then pressing and releasing ALT will bring up the GO-Global TN window's menu, like clicking on the top left corner. If it is disabled, then pressing and releasing ALT will have no effect.

5.9.6 ‘Ensure window is always on top’

If this option is enabled, the GO-Global TN window will stay on top of all other windows.

5.9.7 ‘Full screen on Alt-Enter’

If this option is enabled, then pressing Alt-Enter will cause the GO-Global TN window to become full-screen. Pressing Alt-Enter again will restore the previous window size.

The full-screen feature is also available from the System menu, even when it is configured not to be available on the Alt-Enter key. See section 4.1.3.6.

5.10 The Translation panel

The Translation configuration panel allows you to control the translation between the character set understood by the server and the character set understood by GO-Global TN.

5.10.1 Controlling character set translation

During an interactive session, GO-Global TN receives a stream of 8-bit bytes from the server, and in order to display them on the screen it needs to know what character set to interpret them in.

There are a lot of character sets to choose from. The ‘Received data assumed to be in which character set’ option lets you select one. By default GO-Global TN will attempt to choose a character set that is right for your locale as reported by Windows; if it gets it wrong, you can select a different one using this control.

A few notable character sets are:

- The ISO-8859 series are all standard character sets that include various accented characters appropriate for different sets of languages.
- The Win125x series are defined by Microsoft, for similar purposes. In particular Win1252 is almost equivalent to ISO-8859-1, but contains a few extra characters such as matched quotes and the Euro symbol.
- If you want the old IBM PC character set with block graphics and line-drawing characters, you can select ‘CP437’.
- GO-Global TN also supports Unicode mode, in which the data coming from the server is interpreted as being in the UTF-8 encoding of Unicode. If you select ‘UTF-8’ as a character set you can use this mode. Not all server-side applications will support it.

If you need support for a numeric code page which is not listed in the drop-down list, such as code page 866, then you can try entering its name manually (`CP8766` for example) in the list box. If the underlying version of Windows has the appropriate translation table installed, GO-Global TN will use it.

5.10.2 ‘Caps Lock acts as Cyrillic switch’

This feature allows you to switch between a US/UK keyboard layout and a Cyrillic keyboard layout by using the Caps Lock key, if you need to type (for example) Russian and English side by side in the same document.

Currently this feature is not expected to work properly if your native keyboard layout is not US or UK.

5.10.3 Controlling display of line drawing characters

VT100-series terminals allow the server to send control sequences that shift temporarily into a separate character set for drawing lines and boxes. GO-Global TN has a variety of ways to support this capability. In general you should probably try lots of options until you find one that your particular font supports.

- ‘Font has XWindows encoding’ is for use with fonts that have a special encoding, where the lowest 32 character positions (below the ASCII printable range) contain the line-drawing characters. This is unlikely to be the case with any standard Windows font; it will probably only apply to custom-built fonts or fonts that have been automatically converted from the X Window System.

- ‘Use font in both ANSI and OEM modes’ tries to use the same font in two different character sets, to obtain a wider range of characters. This doesn't always work; some fonts claim to be a different size depending on which character set you try to use.
- ‘Use font in OEM mode only’ is more reliable than that, but can miss out other characters from the main character set.
- ‘Poor man's line drawing’ assumes that the font *cannot* generate the line and box characters at all, so it will use the `+`, `-` and `|` characters to draw approximations to boxes. You should use this option if none of the other options works.
- ‘Unicode mode’ tries to use the box characters that are present in Unicode. For good Unicode-supporting fonts this is probably the most reliable and functional option.

5.10.4 Controlling copy and paste of line drawing characters

By default, when you copy and paste a piece of the GO-Global TN screen that contains VT100 line and box drawing characters, GO-Global TN will paste them in the form they appear on the screen: either Unicode line drawing code points, or the ‘poor man's’ line-drawing characters `+`, `-` and `|`. The checkbox ‘Copy and paste VT100 line drawing characters as lqqqk’ disables this feature, so line-drawing characters will be pasted as the ASCII characters that were printed to produce them. This will typically mean they come out mostly as `q` and `x`, with a scattering of `jklmntuvw` at the corners. This might be useful if you were trying to recreate the same box layout in another program, for example.

Note that this option only applies to line-drawing characters which *were* printed using the VT100 mechanism. Line-drawing characters displayed using Unicode will paste as Unicode always.

5.11 The Selection panel

The selection panel allows you to control the way copy and paste work in the GO-Global TN window.

5.11.1 Pasting in Rich Text Format

If you enable ‘Paste to clipboard in RTF as well as plain text’, GO-Global TN will write formatting information to the clipboard as well as the actual text you copy. Currently the only effect of this will be that if you paste into (say) a word processor, the text will appear in the word processor in the same font GO-Global TN was using to display it. In future it is likely that other formatting information (bold, underline, colours) will be copied as well.

This option can easily be inconvenient, so by default it is disabled.

5.11.2 Changing the actions of the mouse buttons

GO-Global TN's copy and paste mechanism is modelled on the Unix `xterm` application. The X Window System uses a three-button mouse, and the convention is that the left button selects, the right button extends an existing selection, and the middle button pastes.

Windows typically only has two mouse buttons, so in GO-Global TN's default configuration (‘Compromise’), the *right* button pastes, and the *middle* button (if you have one) extends a selection.

If you have a three-button mouse and you are already used to the `xterm` arrangement, you can select it using the ‘Action of mouse buttons’ control.

Alternatively, with the ‘Windows’ option selected, the middle button extends, and the right button brings up a context menu (on which one of the options is ‘Paste’). (This context menu is always available by holding down Ctrl and right-clicking, regardless of the setting of this option.)

5.11.3 ‘Shift overrides application's use of mouse’

GO-Global TN allows the server to send control codes that let it take over the mouse and use it for purposes other than copy and paste. Applications which use this feature include the text-mode web browser `links`, the Usenet newsreader `trn` version 4, and the file manager `mc` (Midnight Commander).

When running one of these applications, pressing the mouse buttons no longer performs copy and paste. If you do need to copy and paste, you can still do so if you hold down Shift while you do your mouse clicks.

However, it is possible in theory for applications to even detect and make use of Shift + mouse clicks. We don't know of any applications that do this, but in case someone ever writes one, unchecking the ‘Shift overrides application's use of mouse’ checkbox will cause Shift + mouse clicks to go to the server as well (so that mouse-driven copy and paste will be completely disabled).

If you want to prevent the application from taking over the mouse at all, you can do this using the Features control panel; see section 5.6.2.

5.11.4 Default selection mode

As described in section 4.1.1, GO-Global TN has two modes of selecting text to be copied to the clipboard. In the default mode (‘Normal’), dragging the mouse from point A to point B selects to the end of the line containing A, all the lines in between, and from the very beginning of the line containing B. In the other mode (‘Rectangular block’), dragging the mouse between two points defines a rectangle, and everything within that rectangle is copied.

Normally, you have to hold down Alt while dragging the mouse to select a rectangular block. Using the ‘Default selection mode’ control, you can set rectangular selection as the default, and then you have to hold down Alt to get the *normal* behaviour.

5.11.5 Configuring word-by-word selection

GO-Global TN will select a word at a time in the terminal window if you double-click to begin the drag. This panel allows you to control precisely what is considered to be a word.

Each character is given a *class*, which is a small number (typically 0, 1 or 2). GO-Global TN considers a single word to be any number of adjacent characters in the same class. So by modifying the assignment of characters to classes, you can modify the word-by-word selection behaviour.

In the default configuration, the character classes are:

- Class 0 contains white space and control characters.
- Class 1 contains most punctuation.
- Class 2 contains letters, numbers and a few pieces of punctuation (the double quote, minus sign, period, forward slash and underscore).

So, for example, if you assign the @ symbol into character class 2, you will be able to select an e-mail address with just a double click.

In order to adjust these assignments, you start by selecting a group of characters in the list box. Then enter a class number in the edit box below, and press the ‘Set’ button.

This mechanism currently only covers ASCII characters, because it isn't feasible to expand the list to cover the whole of Unicode.

Character class definitions can be modified by control sequences sent by the server. This configuration option controls the *default* state, which will be restored when you reset the terminal (see section 4.1.3.5). However, if you modify this option in mid-session using ‘Change Settings’, it will take effect immediately.

5.12 The Colours panel

The Colours panel allows you to control GO-Global TN's use of colour.

5.12.1 ‘Bolded text is a different colour’

When the server sends a control sequence indicating that some text should be displayed in bold, GO-Global TN can handle this two ways. It can either change the font for a bold version, or use the same font in a brighter colour. This control lets you choose which.

By default the box is checked, so non-bold text is displayed in light grey and bold text is displayed in bright white (and similarly in other colours). If you uncheck the box, bold and non-bold text will be displayed in the same colour, and instead the font will change to indicate the difference.

5.12.2 ‘Attempt to use logical palettes’

Logical palettes are a mechanism by which a Windows application running on an 8-bit colour display can select precisely the colours it wants instead of going with the Windows standard defaults.

If you are not getting the colours you ask for on an 8-bit display, you can try enabling this option. However, be warned that it's never worked very well.

5.12.3 ‘Use system colours’

Enabling this option will cause GO-Global TN to ignore the configured colours for ‘Default Background/Foreground’ and ‘Cursor Colour/Text’ (see section 5.12.4), instead going with the system-wide defaults.

Note that non-bold and bold text will be the same colour if this option is enabled. You might want to change to inidicating bold text by font changes (see section 5.12.1).

5.12.4 Adjusting the colours in the terminal window

The main colour control allows you to specify exactly what colours things should be displayed in. To modify one of the GO-Global TN colours, use the list box to select which colour you want to modify. The RGB values for that colour will appear on the right-hand side of the list box. Now, if you press the ‘Modify’ button, you will be presented with a colour selector, in which you can choose a new colour to go in place of the old one.

GO-Global TN allows you to set the cursor colour, the default foreground and background, and the precise shades of all the ANSI configurable colours (black, red, green, yellow, blue, magenta, cyan, and white). You can also modify the precise shades used for the bold versions of these colours; these are used to display bold text if you have selected ‘Bolded text is a different colour’, and can also be used if the server asks specifically to use them.

5.13 The Connection panel

The Connection panel allows you to configure options that apply to more than one type of connection.

5.13.1 ‘Terminal-type string’

Most servers you might connect to with GO-Global TN are designed to be connected to from lots of different types of terminal. In order to send the right control sequences to each one, the server will need to know what type of terminal it is dealing with. Therefore, each of the SSH, Telnet and Rlogin protocols allow a text string to be sent down the connection describing the terminal.

GO-Global TN attempts to emulate the Unix `xterm` program, and by default it reflects this by sending `xterm` as a terminal-type string. If you find this is not doing what you want - perhaps the remote terminal reports ‘Unknown terminal type’ - you could try setting this to something different, such as `vt220`.

If you're not sure whether a problem is due to the terminal type setting or not, you probably need to consult the manual for your application or your server.

5.13.2 ‘Terminal speeds’

The Telnet, Rlogin, and SSH protocols allow the client to specify terminal speeds to the server.

This parameter does *not* affect the actual speed of the connection, which is always ‘as fast as possible’; it is just a hint that is sometimes used by server software to modify its behaviour. For instance, if a slow speed is indicated, the server may switch to a less bandwidth-hungry display mode.

The value is usually meaningless in a network environment, but GO-Global TN lets you configure it, in case you find the server is reacting badly to the default value.

The format is a pair of numbers separated by a comma, for instance, `38400,38400`. The first number represents the output speed (*from* the server) in bits per second, and the second is the input speed (*to* the server). (Only the first is used in the Rlogin protocol.)

This option has no effect on Raw connections.

5.13.3 ‘Auto-login username’

All three of the SSH, Telnet and Rlogin protocols allow you to specify what user name you want to log in as, without having to type it explicitly every time. (Some Telnet servers don't support this.)

In this box you can type that user name.

5.13.4 Using keepalives to prevent disconnection

If you find your sessions are closing unexpectedly (‘Connection reset by peer’) after they have been idle for a while, you might want to try using this option.

Some network routers and firewalls need to keep track of all connections through them. Usually, these firewalls will assume a connection is dead if no data is transferred in either direction after a certain time interval. This can cause GO-Global TN sessions to be unexpectedly closed by the firewall if no traffic is seen in the session for some time.

The keepalive option (‘Seconds between keepalives’) allows you to configure GO-Global TN to send data through the session at regular intervals, in a way that does not disrupt the actual terminal session. If you find your firewall is cutting idle connections off, you can try entering a non-zero value in this field. The value is measured in seconds; so, for example, if your firewall cuts connections off after ten minutes then you might want to enter 300 seconds (5 minutes) in the box.

Note that keepalives are not always helpful. They help if you have a firewall which drops your connection after an idle period; but if the network between you and the server suffers from breaks in connectivity then keepalives can actually make things worse. If a session is idle, and connectivity is temporarily lost between the endpoints, but the connectivity is restored before either side tries to send anything, then there will be no problem - neither endpoint will notice that anything was wrong. However, if one side does send something during the break, it will repeatedly try to re-send, and eventually give up and abandon the connection. Then when connectivity is restored, the other side will find that the first side doesn't believe there is an open connection any more. Keepalives can make this sort of problem worse, because they increase the probability that GO-Global TN will attempt to send data during a break in connectivity. Therefore, you might find they help connection loss, or you might find they make it worse, depending on what *kind* of network problems you have between you and the server.

Keepalives are only supported in Telnet and SSH; the Rlogin and Raw protocols offer no way of implementing them. (For an alternative, see section 5.13.6.)

Note that if you are using SSH1 and the server has a bug that makes it unable to deal with SSH1 ignore messages (see section 5.20.1), enabling keepalives will have no effect.

5.13.5 ‘Disable Nagle's algorithm’

Nagle's algorithm is a detail of TCP/IP implementations that tries to minimise the number of small data packets sent down a network connection. With Nagle's algorithm enabled, GO-Global TN's bandwidth usage will be slightly more efficient; with it disabled, you may find you get a faster response to your keystrokes when connecting to some types of server.

The Nagle algorithm is disabled by default.

5.13.6 ‘Enable TCP keepalives’

NOTE: TCP keepalives should not be confused with the application-level keepalives described in section 5.13.4. If in doubt, you probably want application-level keepalives; TCP keepalives are provided for completeness.

The idea of TCP keepalives is similar to application-level keepalives, and the same caveats apply. The main differences are:

- TCP keepalives are available on *all* connection types, including Raw and Rlogin.
- The interval between TCP keepalives is usually much longer, typically two hours; this is set by the operating system, and cannot be configured within GO-Global TN.
- If the operating system does not receive a response to a keepalive, it may send out more in quick succession and if terminate the connection if no response is received.

TCP keepalives may be more useful for ensuring that half-open connections are terminated than for keeping a connection alive. TCP keepalives are disabled by default.

5.14 The Proxy panel

The Proxy panel allows you to configure GO-Global TN to use various types of proxy in order to make its network connections. The settings in this panel affect the primary network connection forming your GO-Global TN session, but also any extra connections made as a result of SSH port forwarding (see section 4.5).

5.14.1 Setting the proxy type

The ‘Proxy type’ radio buttons allow you to configure what type of proxy you want GO-Global TN to use for its network connections. The default setting is ‘None’; in this mode no proxy is used for any connection.

- Selecting ‘HTTP’ allows you to proxy your connections through a web server supporting the HTTP `CONNECT` command, as documented in RFC 2817.
- Selecting ‘SOCKS 4’ or ‘SOCKS 5’ allows you to proxy your connections through a SOCKS server.
- Many firewalls implement a less formal type of proxy in which a user can make a Telnet connection directly to the firewall machine and enter a command such as `connect myhost.com 22` to connect through to an external host. Selecting ‘Telnet’ allows you to tell GO-Global TN to use this type of proxy.

5.14.2 Excluding parts of the network from proxying

Typically you will only need to use a proxy to connect to non-local parts of your network; for example, your proxy might be required for connections outside your company's internal network. In the ‘Exclude Hosts/IPs’ box you can enter ranges of IP addresses, or ranges of DNS names, for which GO-Global TN will avoid using the proxy and make a direct connection instead.

The ‘Exclude Hosts/IPs’ box may contain more than one exclusion range, separated by commas. Each range can be an IP address or a DNS name, with a `*` character allowing wildcards. For example:

```
*.example.com
```

This excludes any host with a name ending in `.example.com` from proxying.

```
192.168.88.*
```

This excludes any host with an IP address starting with 192.168.88 from proxying.

```
192.168.88.*,*.example.com
```

This excludes both of the above ranges at once.

Connections to the local host (the host name `localhost`, and any loopback IP address) are never proxied, even if the proxy exclude list does not explicitly contain them. It is very unlikely that this behaviour would ever cause problems, but if it does you can change it by enabling 'Consider proxying local host connections'.

Note that if you are doing DNS at the proxy (see section 5.14.3), you should make sure that your proxy exclusion settings do not depend on knowing the IP address of a host. If the name is passed on to the proxy without GO-Global TN looking it up, it will never know the IP address and cannot check it against your list.

5.14.3 Name resolution when using a proxy

If you are using a proxy to access a private network, it can make a difference whether DNS name resolution is performed by GO-Global TN itself (on the client machine) or performed by the proxy.

The 'Do DNS name lookup at proxy end' configuration option allows you to control this. If you set it to 'No', GO-Global TN will always do its own DNS, and will always pass an IP address to the proxy. If you set it to 'Yes', GO-Global TN will always pass host names straight to the proxy without trying to look them up first.

If you set this option to 'Auto' (the default), GO-Global TN will do something it considers appropriate for each type of proxy. Telnet and HTTP proxies will have host names passed straight to them; SOCKS proxies will not.

Note that if you are doing DNS at the proxy, you should make sure that your proxy exclusion settings (see section 5.14.2) do not depend on knowing the IP address of a host. If the name is passed on to the proxy without GO-Global TN looking it up, it will never know the IP address and cannot check it against your list.

The original SOCKS 4 protocol does not support proxy-side DNS. There is a protocol extension (SOCKS 4A) which does support it, but not all SOCKS 4 servers provide this extension. If you enable proxy DNS and your SOCKS 4 server cannot deal with it, this might be why.

5.14.4 Username and password

If your proxy requires authentication, you can enter a username and a password in the 'Username' and 'Password' boxes.

Note that if you save your session, the proxy password will be saved in plain text, so anyone who can access your GO-Global TN configuration data will be able to discover it.

Authentication is not fully supported for all forms of proxy:

- Username and password authentication is supported for HTTP proxies and SOCKS 5 proxies.
- SOCKS 4 can use the ‘Username’ field, but does not support passwords.
- You can specify a way to include a username and password in the Telnet proxy command (see section 5.14.5).

5.14.5 Specifying the Telnet proxy command

If you are using the Telnet proxy type, the usual command required by the firewall's Telnet server is `connect`, followed by a host name and a port number. If your proxy needs a different command, you can enter an alternative here.

In this string, you can use `\n` to represent a new-line, `\r` to represent a carriage return, `\t` to represent a tab character, and `\x` followed by two hex digits to represent any other character. `\\` is used to encode the `\` character itself.

Also, the special strings `%host` and `%port` will be replaced by the host name and port number you want to connect to. The strings `%user` and `%pass` will be replaced by the proxy username and password you specify. To get a literal `%` sign, enter `%%`.

If the Telnet proxy server prompts for a username and password before commands can be sent, you can use a command such as:

```
%user\n%pass\nconnect %host %port\n
```

This will send your username and password as the first two lines to the proxy, followed by a command to connect to the desired host and port. Note that if you do not include the `%user` or `%pass` tokens in the Telnet command, then the ‘Username’ and ‘Password’ configuration fields will be ignored.

5.15 The Telnet panel

The Telnet panel allows you to configure options that only apply to Telnet sessions.

5.15.1 Setting environment variables on the server

The Telnet protocol provides a means for the client to pass environment variables to the server. Many Telnet servers have stopped supporting this feature due to security flaws, but GO-Global TN still supports it for the benefit of any servers which have found other ways around the security problems than just disabling the whole mechanism.

To add an environment variable to the list transmitted down the connection, you enter the variable name in the ‘Variable’ box, enter its value in the ‘Value’ box, and press the ‘Add’ button. To remove one from the list, select it in the list box and press ‘Remove’.

5.15.2 ‘Handling of OLD_ENVIRON ambiguity’

The original Telnet mechanism for passing environment variables was badly specified. At the time the standard (RFC 1408) was written, BSD telnet implementations were already supporting the feature, and the intention of the standard was to describe the behaviour the BSD implementations were already using.

Sadly there was a typing error in the standard when it was issued, and two vital function codes were specified the wrong way round. BSD implementations did not change, and the standard was not corrected. Therefore, it's possible you might find either BSD or RFC-compliant implementations out there. This switch allows you to choose which one GO-Global TN claims to be.

The problem was solved by issuing a second standard, defining a new Telnet mechanism called `NEW_ENVIRON`, which behaved exactly like the original `OLD_ENVIRON` but was not encumbered by existing implementations. Most Telnet servers now support this, and it's unambiguous. This feature should only be needed if you have trouble passing environment variables to quite an old server.

5.15.3 Passive and active Telnet negotiation modes

In a Telnet connection, there are two types of data passed between the client and the server: actual text, and *negotiations* about which Telnet extra features to use.

GO-Global TN can use two different strategies for negotiation:

- In *active* mode, GO-Global TN starts to send negotiations as soon as the connection is opened.
- In *passive* mode, GO-Global TN will wait to negotiate until it sees a negotiation from the server.

The obvious disadvantage of passive mode is that if the server is also operating in a passive mode, then negotiation will never begin at all. For this reason GO-Global TN defaults to active mode.

However, sometimes passive mode is required in order to successfully get through certain types of firewall and Telnet proxy server. If you have confusing trouble with a firewall, you could try enabling passive mode to see if it helps.

5.15.4 ‘Keyboard sends telnet Backspace and Interrupt’

If this box is checked, the Backspace key on the keyboard will send the Telnet special backspace code, and Control-C will send the Telnet special interrupt code. You probably shouldn't enable this unless you know what you're doing.

5.15.5 ‘Return key sends telnet New Line instead of ^M’

Unlike most other remote login protocols, the Telnet protocol has a special ‘new line’ code that is not the same as the usual line endings of Control-M or Control-J. By default, GO-Global TN sends the Telnet New Line code when you press Return, instead of sending Control-M as it does in most other protocols.

Most Unix-style Telnet servers don't mind whether they receive Telnet New Line or Control-M; some servers do expect New Line, and some servers prefer to see ^M. If you are seeing surprising behaviour when you press Return in a Telnet session, you might try turning this option off to see if it helps.

5.16 The Rlogin panel

The Rlogin panel allows you to configure options that only apply to Rlogin sessions.

5.16.1 'Local username'

Rlogin allows an automated (password-free) form of login by means of a file called `.rhosts` on the server. You put a line in your `.rhosts` file saying something like `jbloggs@pc1.example.com`, and then when you make an Rlogin connection the client transmits the username of the user running the Rlogin client. The server checks the username and hostname against `.rhosts`, and if they match it does not ask for a password.

This only works because Unix systems contain a safeguard to stop a user from pretending to be another user in an Rlogin connection. Rlogin connections have to come from port numbers below 1024, and Unix systems prohibit this to unprivileged processes; so when the server sees a connection from a low-numbered port, it assumes the client end of the connection is held by a privileged (and therefore trusted) process, so it believes the claim of who the user is.

Windows does not have this restriction: *any* user can initiate an outgoing connection from a low-numbered port. Hence, the Rlogin `.rhosts` mechanism is completely useless for securely distinguishing several different users on a Windows machine. If you have a `.rhosts` entry pointing at a Windows PC, you should assume that *anyone* using that PC can spoof your username in an Rlogin connection and access your account on the server.

The 'Local username' control allows you to specify what user name GO-Global TN should claim you have, in case it doesn't match your Windows user name (or in case you didn't bother to set up a Windows user name).

5.17 The SSH panel

The SSH panel allows you to configure options that only apply to SSH sessions.

5.17.1 Executing a specific command on the server

In SSH, you don't have to run a general shell session on the server. Instead, you can choose to run a single specific command (such as a mail user agent, for example). If you want to do this, enter the command in the 'Remote command' box.

5.17.2 'Don't allocate a pseudo-terminal'

When connecting to a Unix system, most interactive shell sessions are run in a *pseudo-terminal*, which allows the Unix system to pretend it's talking to a real physical terminal device but allows the SSH server to catch all the data coming from that fake device and send it back to the client.

Occasionally you might find you have a need to run a session *not* in a pseudo-terminal. In GO-Global TN, this is generally only useful for very specialist purposes.

5.17.3 'Enable compression'

This enables data compression in the SSH connection: data sent by the server is compressed before sending, and decompressed at the client end. Likewise, data sent by GO-Global TN to the server is compressed first and the server decompresses it at the other end. This can help make the most of a low-bandwidth connection.

5.17.4 ‘Preferred SSH protocol version’

This allows you to select whether you would like to use SSH protocol version 1 or version 2.

GO-Global TN will attempt to use protocol 1 if the server you connect to does not offer protocol 2, and vice versa.

If you select ‘1 only’ or ‘2 only’ here, GO-Global TN will only connect if the server you connect to offers the SSH protocol version you have specified.

5.17.5 Encryption algorithm selection

GO-Global TN supports a variety of different encryption algorithms, and allows you to choose which one you prefer to use. You can do this by dragging the algorithms up and down in the list box (or moving them using the Up and Down buttons) to specify a preference order. When you make an SSH connection, GO-Global TN will search down the list from the top until it finds an algorithm supported by the server, and then use that.

GO-Global TN currently supports the following algorithms:

- AES (Rijndael) - 256, 192, or 128-bit CBC (SSH-2 only)
- Blowfish - 128-bit CBC
- Triple-DES - 168-bit CBC
- Single-DES - 56-bit CBC (see below for SSH-2)

If the algorithm GO-Global TN finds is below the ‘warn below here’ line, you will see a warning box when you make the connection:

```
The first cipher supported by the server
is single-DES, which is below the configured
warning threshold.
Do you want to continue with this connection?
```

This warns you that the first available encryption is not a very secure one. Typically you would put the ‘warn below here’ line between the encryptions you consider secure and the ones you consider substandard. By default, GO-Global TN supplies a preference order intended to reflect a reasonable preference in terms of security and speed.

In SSH-2, the encryption algorithm is negotiated independently for each direction of the connection, although GO-Global TN does not support separate configuration of the preference orders. As a result you may get two warnings similar to the one above, possibly with different encryptions.

Single-DES is not recommended in the SSH 2 draft protocol standards, but one or two server implementations do support it. GO-Global TN can use single-DES to interoperate with these servers if you enable the ‘Enable legacy use of single-DES in SSH 2’ option; by default this is disabled and GO-Global TN will stick to recommended ciphers.

5.18 The Auth panel

The Auth panel allows you to configure authentication options for SSH sessions.

5.18.1 ‘Attempt TIS or CryptoCard authentication’

TIS and CryptoCard authentication are simple challenge/response forms of authentication available in SSH protocol version 1 only. You might use them if you were using S/Key one-time passwords, for example, or if you had a physical security token that generated responses to authentication challenges.

With this switch enabled, GO-Global TN will attempt these forms of authentication if the server is willing to try them. You will be presented with a challenge string (which will be different every time) and must supply the correct response in order to log in. If your server supports this, you should talk to your system administrator about precisely what form these challenges and responses take.

5.18.2 ‘Attempt keyboard-interactive authentication’

The SSH 2 equivalent of TIS authentication is called ‘keyboard-interactive’. It is a flexible authentication method using an arbitrary sequence of requests and responses; so it is not only useful for challenge/response mechanisms such as S/Key, but it can also be used for (for example) asking the user for a new password when the old one has expired.

GO-Global TN leaves this option enabled by default, but supplies a switch to turn it off in case you should have trouble with it.

5.18.3 ‘Allow agent forwarding’

This option allows the SSH server to open forwarded connections back to your local copy of Pageant. If you are not running Pageant, this option will do nothing.

See chapter 7 for general information on Pageant, and section 7.4 for information on agent forwarding. Note that there is a security risk involved with enabling this option; see section 7.5 for details.

5.18.4 ‘Allow attempted changes of username in SSH2’

In the SSH 1 protocol, it is impossible to change username after failing to authenticate. So if you mis-type your username at the GO-Global TN ‘login as:’ prompt, you will not be able to change it except by restarting GO-Global TN.

The SSH 2 protocol *does* allow changes of username, in principle, but does not make it mandatory for SSH 2 servers to accept them. In particular, OpenSSH does not accept a change of username; once you have sent one username, it will reject attempts to try to authenticate as another user. (Depending on the version of OpenSSH, it may quietly return failure for all login attempts, or it may send an error message.)

For this reason, GO-Global TN will by default not prompt you for your username more than once, in case the server complains. If you know your server can cope with it, you can enable the ‘Allow attempted changes of username’ option to modify GO-Global TN's behaviour.

5.18.5 ‘Private key file for authentication’

This box is where you enter the name of your private key file if you are using public key authentication. See chapter 6 for information about public key authentication in SSH.

This key must be in GO-Global TN's native format (`*.PPK`).

5.19 The Tunnels panel

The Tunnels panel allows you to configure tunnelling of other connection types through an SSH connection.

5.19.1 X11 forwarding

If your server lets you run X Window System applications, X11 forwarding allows you to securely give those applications access to a local X display on your PC.

To enable X11 forwarding, check the ‘Enable X11 forwarding’ box. If your X display is not the primary display on your local machine (which it almost certainly will be unless you have deliberately arranged otherwise), you need to enter its location in the ‘X display location’ box.

See section 4.4 for more information about X11 forwarding.

5.19.1.1 Remote X11 authentication

If you are using X11 forwarding, the virtual X server created on the SSH server machine will be protected by authorisation data. This data is invented, and checked, by GO-Global TN.

The usual authorisation method used for this is called `MIT-MAGIC-COOKIE-1`. This is a simple password-style protocol: the X client sends some cookie data to the server, and the server checks that it matches the real cookie. The cookie data is sent over an unencrypted X11 connection; so if you allow a client on a third machine to access the virtual X server, then the cookie will be sent in the clear.

GO-Global TN offers the alternative protocol `XDM-AUTHORIZATION-1`. This is a cryptographically authenticated protocol: the data sent by the X client is different every time, and it depends on the IP address and port of the client's end of the connection and is also stamped with the current time. So an eavesdropper who captures an `XDM-AUTHORIZATION-1` string cannot immediately re-use it for their own X connection.

GO-Global TN's support for `XDM-AUTHORIZATION-1` is a somewhat experimental feature, and may encounter several problems:

- Some X clients probably do not even support `XDM-AUTHORIZATION-1`, so they will not know what to do with the data GO-Global TN has provided.
- This authentication mechanism will only work in SSH v2. In SSH v1, the SSH server does not tell the client the source address of a forwarded connection in a machine-readable format, so it's impossible to verify the `XDM-AUTHORIZATION-1` data.
- You may find this feature causes problems with some SSH servers, which will not clean up `XDM-AUTHORIZATION-1` data after a session, so that if you then connect to the same server using a client which only does `MIT-MAGIC-COOKIE-1` and are allocated the same remote display number, you might find that out-of-date authentication data is still present on your server and your X connections fail.

GO-Global TN's default is `MIT-MAGIC-COOKIE-1`. If you change it, you should be sure you know what you're doing.

5.19.2 Port forwarding

Port forwarding allows you to tunnel other types of network connection down an SSH session. See section 4.5 for a general discussion of port forwarding and how it works.

The port forwarding section in the Tunnels panel shows a list of all the port forwardings that GO-Global TN will try to set up when it connects to the server. By default no port forwardings are set up, so this list is empty.

To add a port forwarding:

- Set one of the ‘Local’ or ‘Remote’ radio buttons, depending on whether you want to forward a local port to a remote destination (‘Local’) or forward a remote port to a local destination (‘Remote’). Alternatively, select ‘Dynamic’ if you want GO-Global TN to provide a local SOCKS 4/4A/5 proxy on a local port.
- Enter a source port number into the ‘Source port’ box. For local forwardings, GO-Global TN will listen on this port of your PC. For remote forwardings, your SSH server will listen on this port of the remote machine. Note that most servers will not allow you to listen on port numbers less than 1024.
- If you have selected ‘Local’ or ‘Remote’ (this step is not needed with ‘Dynamic’), enter a hostname and port number separated by a colon, in the ‘Destination’ box. Connections received on the source port will be directed to this destination. For example, to connect to a POP-3 server, you might enter `popserver.example.com:110`.
- Click the ‘Add’ button. Your forwarding details should appear in the list box.

To remove a port forwarding, simply select its details in the list box, and click the ‘Remove’ button.

In the ‘Source port’ box, you can also optionally enter an IP address to listen on, by specifying (for instance) `127.0.0.5:79`. See section 4.5 for more information on how this works and its restrictions.

5.19.3 Controlling the visibility of forwarded ports

The source port for a forwarded connection usually does not accept connections from any machine except the SSH client or server machine itself (for local and remote forwardings respectively). There are controls in the Tunnels panel to change this:

- The ‘Local ports accept connections from other hosts’ option allows you to set up local-to-remote port forwardings in such a way that machines other than your client PC can connect to the forwarded port. (This also applies to dynamic SOCKS forwarding.)
- The ‘Remote ports do the same’ option does the same thing for remote-to-local port forwardings (so that machines other than the SSH server machine can connect to the forwarded port.) Note that this feature is only available in the SSH 2 protocol, and not all SSH 2 servers support it (OpenSSH 3.0 does not, for example).

5.20 The Bugs panel

Not all SSH servers work properly. Various existing servers have bugs in them, which can make it impossible for a client to talk to them unless it knows about the bug and works around it.

Since most servers announce their software version number at the beginning of the SSH connection, GO-Global TN will attempt to detect which bugs it can expect to see in the server and automatically enable workarounds. However, sometimes it will make mistakes; if the server has been deliberately configured to conceal its version number, or if the server is a version which GO-Global TN's bug database does not know about, then GO-Global TN will not know what bugs to expect.

The Bugs panel allows you to manually configure the bugs GO-Global TN expects to see in the server. Each bug can be configured in three states:

- ‘Off’: GO-Global TN will assume the server does not have the bug.
- ‘On’: GO-Global TN will assume the server *does* have the bug.
- ‘Auto’: GO-Global TN will use the server's version number announcement to try to guess whether or not the server has the bug.

5.20.1 ‘Chokes on SSH1 ignore messages’

An ignore message (SSH_MSG_IGNORE) is a message in the SSH protocol which can be sent from the client to the server, or from the server to the client, at any time. Either side is required to ignore the message whenever it receives it. GO-Global TN uses ignore messages to hide the password packet in SSH1, so that a listener cannot tell the length of the user's password; it also uses ignore messages for connection keepalives (see section 5.13.4).

If this bug is detected, GO-Global TN will stop using ignore messages. This means that keepalives will stop working, and GO-Global TN will have to fall back to a secondary defence against SSH1 password-length eavesdropping. See section 5.20.2. If this bug is enabled when talking to a correct server, the session will succeed, but keepalives will not work and the session might be more vulnerable to eavesdroppers than it could be.

This is an SSH1-specific bug. No known SSH2 server fails to deal with SSH2 ignore messages.

5.20.2 ‘Refuses all SSH1 password camouflage’

When talking to an SSH1 server which cannot deal with ignore messages (see section 5.20.1), GO-Global TN will attempt to disguise the length of the user's password by sending additional padding *within* the password packet. This is technically a violation of the SSH1 specification, and so GO-Global TN will only do it when it cannot use standards-compliant ignore messages as camouflage. In this sense, for a server to refuse to accept a padded password packet is not really a bug, but it does make life inconvenient if the server can also not handle ignore messages.

If this ‘bug’ is detected, GO-Global TN will have no choice but to send the user's password with no form of camouflage, so that an eavesdropping user will be easily able to find out the exact length of the password. If this bug is enabled when talking to a correct server, the session will succeed, but will be more vulnerable to eavesdroppers than it could be.

This is an SSH1-specific bug. SSH2 is secure against this type of attack.

5.20.3 ‘Chokes on SSH1 RSA authentication’

Some SSH1 servers cannot deal with RSA authentication messages at all. If GGTNAuth is running and contains any SSH1 keys, GO-Global TN will normally automatically try RSA authentication before falling back to passwords, so these servers will crash when they see the RSA attempt.

If this bug is detected, GO-Global TN will go straight to password authentication. If this bug is enabled when talking to a correct server, the session will succeed, but of course RSA authentication will be impossible.

This is an SSH1-specific bug.

5.20.4 'Miscomputes SSH2 HMAC keys'

Versions 2.3.0 and below of the SSH server software from `ssh.com` compute the keys for their HMAC message authentication codes incorrectly. A typical symptom of this problem is that GO-Global TN dies unexpectedly at the beginning of the session, saying 'Incorrect MAC received on packet'.

If this bug is detected, GO-Global TN will compute its HMAC keys in the same way as the buggy server, so that communication will still be possible. If this bug is enabled when talking to a correct server, communication will fail.

This is an SSH2-specific bug.

5.20.5 'Miscomputes SSH2 encryption keys'

Versions below 2.0.11 of the SSH server software from `ssh.com` compute the keys for the session encryption incorrectly. This problem can cause various error messages, such as 'Incoming packet was garbled on decryption', or possibly even 'Out of memory'.

If this bug is detected, GO-Global TN will compute its encryption keys in the same way as the buggy server, so that communication will still be possible. If this bug is enabled when talking to a correct server, communication will fail.

This is an SSH2-specific bug.

5.20.6 'Requires padding on SSH2 RSA signatures'

Versions below 3.3 of OpenSSH require SSH2 RSA signatures to be padded with zero bytes to the same length as the RSA key modulus. The SSH2 draft specification says that an unpadded signature MUST be accepted, so this is a bug. A typical symptom of this problem is that GO-Global TN mysteriously fails RSA authentication once in every few hundred attempts, and falls back to passwords.

If this bug is detected, GO-Global TN will pad its signatures in the way OpenSSH expects. If this bug is enabled when talking to a correct server, it is likely that no damage will be done, since correct servers usually still accept padded signatures because they're used to talking to OpenSSH.

This is an SSH2-specific bug.

5.20.7 'Chokes on Diffie-Hellman group exchange'

We have anecdotal evidence that some SSH servers claim to be able to perform Diffie-Hellman group exchange, but fail to actually do so when GO-Global TN tries to. If your SSH2 sessions spontaneously close immediately after opening the GO-Global TN window, it might be worth enabling the workaround for this bug to see if it helps.

We have no hard evidence that any specific version of specific server software reliably demonstrates this bug. Therefore, GO-Global TN will never *assume* a server has this bug; if you want the workaround, you need to enable it manually.

This is an SSH2-specific bug.

5.20.8 'Misuses the session ID in PK auth'

Versions below 2.3 of OpenSSH require SSH2 public-key authentication to be done slightly differently: the data to be signed by the client contains the session ID formatted in a different way. If public-key authentication mysteriously does not work but the Event Log (see section 4.1.3.1) thinks it has successfully sent a signature, it might be worth enabling the workaround for this bug to see if it helps.

If this bug is detected, GO-Global TN will sign data in the way OpenSSH expects. If this bug is enabled when talking to a correct server, SSH2 public-key authentication will fail.

This is an SSH2-specific bug.

5.21 Storing configuration in a file

GO-Global TN does not currently support storing its configuration in a file instead of the Registry. However, you can work around this with a couple of batch files.

You will need a file called (say) `GGTN.BAT` which imports the contents of a file into the Registry, then runs GO-Global TN, exports the contents of the Registry back into the file, and deletes the Registry entries. This can all be done using the Regedit command line options, so it's all automatic. Here is what you need in `GGTN.BAT`:

```
@ECHO OFF
regedit /s ggtn.reg
regedit /s ggtnrnd.reg
start /w ggtn.exe
regedit /e ggtnnew.reg HKEY_CURRENT_USER\Software\Graphon\GO-Global TN
copy ggtnnew.reg ggtn.reg
del ggtnnew.reg
regedit /s ggtndel.reg
```

This batch file needs two auxiliary files: `GGTNRND.REG` which sets up an initial safe location for the `GGTN.RND` random seed file, and `GGTNDEL.REG` which destroys everything in the Registry once it's been successfully saved back to the file.

Here is `GGTNDEL.REG`:

```
REGEDIT4

[-HKEY_CURRENT_USER\Software\GraphOn\GO-Global TN]
```

Here is an example `GGTNRND.REG` file:

```
REGEDIT4

[HKEY_CURRENT_USER\Software\GraphOn\GO-Global TN]
"RandSeedFile"="a:\ggtn.rnd"
```

You should replace `a:\ggtn.rnd` with the location where you want to store your random number data. If the aim is to carry around GO-Global TN and its settings on one floppy, you probably want to store it on the floppy.

Chapter 6: Using public keys for SSH authentication

6.1 Public key authentication - an introduction

Public key authentication is an alternative means of identifying yourself to a login server, instead of typing a password. It is more secure and more flexible, but more difficult to set up.

In conventional password authentication, you prove you are who you claim to be by proving that you know the correct password. The only way to prove you know the password is to tell the server what you think the password is. This means that if the server has been hacked, or *spoofed* (see section 2.2), an attacker can learn your password.

Public key authentication solves this problem. You generate a *key pair*, consisting of a public key (which everybody is allowed to know) and a private key (which you keep secret and do not give to anybody). The private key is able to generate *signatures*. A signature created using your private key cannot be forged by anybody who does not have that key; but anybody who has your public key can verify that a particular signature is genuine.

So you generate a key pair on your own computer, and you copy the public key to the server. Then, when the server asks you to prove who you are, GO-Global TN can generate a signature using your private key. The server can verify that signature (since it has your public key) and allow you to log in. Now if the server is hacked or spoofed, the attacker does not gain your private key or password; they only gain one signature. And signatures cannot be re-used, so they have gained nothing.

There is a problem with this: if your private key is stored unprotected on your own computer, then anybody who gains access to *that* will be able to generate signatures as if they were you. So they will be able to log in to your server under your account. For this reason, your private key is usually *encrypted* when it is stored on your local machine, using a passphrase of your choice. In order to generate a signature, GO-Global TN must decrypt the key, so you have to type your passphrase.

This can make public-key authentication less convenient than password authentication: every time you log in to the server, instead of typing a short password, you have to type a longer passphrase. One solution to this is to use an *authentication agent*, a separate program which holds decrypted private keys and generates signatures on request. GO-Global TN's authentication agent is called GO-Global TN Auth. When you begin a Windows session, you start GO-Global TN Auth and load your public key into it (typing your passphrase once). For the rest of your session, you can start GO-Global TN any number of times and GO-Global TN Auth will automatically generate signatures without you having to do anything. When you close your Windows session, GO-Global TN Auth shuts down, without ever having stored your decrypted private key on disk. Many people feel this is a good compromise between security and convenience. See chapter 7 for further details.

There is more than one public-key algorithm available. The most common is RSA, but others exist, notably DSA (otherwise known as DSS), the USA's federal Digital Signature Standard. The key types supported by GO-Global TN are described in section 6.2.2.

6.2 Using GO-Global TN Keygen, the GO-Global TN key generator

GO-Global TN Keygen is a key generator. It generates pairs of public and private keys to be used with GO-Global TN and with the GO-Global TN authentication agent, GGTN Auth (see chapter 7). GO-Global TN Keygen generates RSA keys.

When you run GO-Global TN Keygen you will see a window where you have two choices: ‘Generate’, to generate a new public/private key pair, or ‘Load’ to load in an existing private key.

6.2.1 Generating a new key

This is a general outline of the procedure for generating a new key pair. The following sections describe the process in more detail.

- First, you need to select which type of key you want to generate, and also select the strength of the key. This is described in more detail in section 6.2.2 and section 6.2.3.
- Then press the ‘Generate’ button, to actually generate the key. Section 6.2.4 describes this step.
- Once you have generated the key, select a comment field (section 6.2.6) and a passphrase (section 6.2.7).
- Now you're ready to save the private key to disk; press the ‘Save private key’ button. (See section 6.2.8).

Your key pair is now ready for use. You may also want to copy the public key to your server, either by copying it out of the ‘Public key for pasting into authorized_keys file’ box (see section 6.2.10), or by using the ‘Save public key’ button (section 6.2.9). However, you don't need to do this immediately; if you want, you can load the private key back into GO-Global TN KeYgen later (see section 6.2.11) and the public key will be available for copying and pasting again.

section 6.3 describes the typical process of configuring GO-Global TN to attempt public-key authentication, and configuring your SSH server to accept it.

6.2.2 Selecting the type of key

Before generating a public key using GO-Global TN Keygen, you need to select which type of key you need. GO-Global TN Keygen currently supports three types of key:

- An RSA key for use with the SSH 1 protocol.
- An RSA key for use with the SSH 2 protocol.
- A DSA key for use with the SSH 2 protocol.

The SSH 1 protocol only supports RSA keys; if you will be connecting using the SSH 1 protocol, you must select the first key type or your key will be completely useless.

The SSH 2 protocol supports more than one key type. The two types supported by GO-Global TN are RSA and DSA.

The GO-Global TN developers *strongly* recommend you use RSA. DSA has an intrinsic weakness which makes it very easy to create a signature which contains enough information to give away the *private* key! This would allow an attacker to pretend to be you for any number of future sessions. GO-Global TN's implementation has taken very careful precautions to avoid this weakness, but we cannot be 100% certain we have managed it, and if you have the choice we strongly recommend using RSA keys instead.

If you really need to connect to an SSH server which only supports DSA, then you probably have no choice but to use DSA. If you do use DSA, we recommend you do not use the same key to authenticate with more than one server.

6.2.3 Selecting the size (strength) of the key

The ‘Number of bits’ input box allows you to choose the strength of the key GO-Global TN Keygen will generate.

Currently 1024 bits should be sufficient for most purposes.

6.2.4 The ‘Generate’ button

Once you have chosen the type of key you want, and the strength of the key, press the ‘Generate’ button and GO-Global TN Keygen will begin the process of actually generating the key.

First, a progress bar will appear and GO-Global TN Keygen will ask you to move the mouse around to generate randomness. Wave the mouse in circles over the blank area in the GO-Global TN Keygen window, and the progress bar will gradually fill up as GO-Global TN Keygen collects enough randomness. You don't need to wave the mouse in particularly imaginative patterns (although it can't hurt); GO-Global TN Keygen will collect enough randomness just from the fine detail of *exactly* how far the mouse has moved each time Windows samples its position.

When the progress bar reaches the end, GO-Global TN Keygen will begin creating the key. The progress bar will reset to the start, and gradually move up again to track the progress of the key generation. It will not move evenly, and may occasionally slow down to a stop; this is unfortunately unavoidable, because key generation is a random process and it is impossible to reliably predict how long it will take.

When the key generation is complete, a new set of controls will appear in the window to indicate this.

6.2.5 The ‘Key fingerprint’ box

The ‘Key fingerprint’ box shows you a fingerprint value for the generated key. This is derived cryptographically from the *public* key value, so it doesn't need to be kept secret.

The fingerprint value is intended to be cryptographically secure, in the sense that it is computationally infeasible for someone to invent a second key with the same fingerprint, or to find a key with a particular fingerprint. So some utilities, such as the GO-Global TN Auth key list box (see section 7.2.1) and the Unix `ssh-add` utility, will list key fingerprints rather than the whole public key.

6.2.6 Setting a comment for your key

If you have more than one key and use them for different purposes, you don't need to memorise the key fingerprints in order to tell them apart. GO-Global TN allows you to enter a *comment* for your key, which will be displayed whenever GO-Global TN or GO-Global TN Auth asks you for the passphrase.

The default comment format, if you don't specify one, contains the key type and the date of generation, such as `rsa-key-20011212`. Another commonly used approach is to use your name and the name of the computer the key will be used on, such as `me@my-pc`.

To alter the key comment, just type your comment text into the ‘Key comment’ box before saving the private key. If you want to change the comment later, you can load the private key back into GO-Global TN Keygen, change the comment, and save it again.

6.2.7 Setting a passphrase for your key

The ‘Key passphrase’ and ‘Confirm passphrase’ boxes allow you to choose a passphrase for your key. The passphrase will be used to encrypt the key on disk, so you will not be able to use the key without first entering the passphrase.

When you save the key, GO-Global TN will check that the ‘Key passphrase’ and ‘Confirm passphrase’ boxes both contain exactly the same passphrase, and will refuse to save the key otherwise.

If you leave the passphrase fields blank, the key will be saved unencrypted. You should *not* do this without good reason; if you do, your private key file on disk will be all an attacker needs to gain access to any machine configured to accept that key. If you want to be able to log in without having to type a passphrase every time, you should consider using GO-Global TN Auth (chapter 7) so that your decrypted key is only held in memory rather than on disk.

Under special circumstances you may genuinely *need* to use a key with no passphrase; for example, if you need to run an automated batch script that needs to make an SSH connection, you can't be there to type the passphrase. In this case we recommend you generate a special key for each specific batch script (or whatever) that needs one, and on the server side you should arrange that each key is *restricted* so that it can only be used for that specific purpose. The documentation for your SSH server should explain how to do this (it will probably vary between servers).

Choosing a good passphrase is difficult. Just as you shouldn't use a dictionary word as a password because it's easy for an attacker to run through a whole dictionary, you should not use a song lyric, quotation or other well-known sentence as a passphrase. DiceWare (`www.diceware.com`) recommends using at least five words each generated randomly by rolling five dice, which gives over 2^64 possible passphrases and is probably not a bad scheme. If you want your passphrase to make grammatical sense, this cuts down the possibilities a lot and you should use a longer one as a result.

Do not forget your passphrase. There is no way to recover it.

6.2.8 Saving your private key to a disk file

Once you have generated a key, set a comment field and set a passphrase, you are ready to save your private key to disk.

Press the ‘Save private key’ button. GO-Global TN Keygen will put up a dialog box asking you where to save the file. Select a directory, type in a file name, and press ‘Save’.

This file is the one you will need to tell GO-Global TN to use for authentication (see section 5.18.5) or tell GO-Global TN Auth to load (see section 7.2.2).

6.2.9 Saving your public key to a disk file

The SSH 2 protocol drafts specify a standard format for storing public keys on disk. Some SSH servers (such as `ssh.com`'s) require a public key in this format in order to accept authentication with the corresponding private key. (Others, such as OpenSSH, use a different format; see section 6.2.10.)

To save your public key in the SSH 2 standard format, press the ‘Save public key’ button in GO-Global TN Keygen. GO-Global TN Keygen will put up a dialog box asking you where to save the file. Select a directory, type in a file name, and press ‘Save’.

You will then probably want to copy the public key file to your SSH server machine. See section 6.3 for general instructions on configuring public-key authentication once you have generated a key.

If you use this option with an SSH 1 key, the file GO-Global TN Keygen saves will contain exactly the same text that appears in the ‘Public key for pasting’ box. This is the only existing standard for SSH 1 public keys.

6.2.10 ‘Public key for pasting into authorized_keys file’

All SSH 1 servers require your public key to be given to it in a one-line format before it will accept authentication with your private key. The OpenSSH server also requires this for SSH 2.

The ‘Public key for pasting into authorized_keys file’ gives the public-key data in the correct one-line format. Typically you will want to select the entire contents of the box using the mouse, press Ctrl+C to copy it to the clipboard, and then paste the data into a GO-Global TN session which is already connected to the server.

See section 6.3 for general instructions on configuring public-key authentication once you have generated a key.

6.2.11 Reloading a private key

GO-Global TN Keygen allows you to load an existing private key file into memory. If you do this, you can then change the passphrase and comment before saving it again; you can also make extra copies of the public key.

To load an existing key, press the ‘Load’ button. GO-Global TN Keygen will put up a dialog box where you can browse around the file system and find your key file. Once you select the file, GO-Global TN Keygen will ask you for a passphrase (if necessary) and will then display the key details in the same way as if it had just generated the key.

6.3 Getting ready for public key authentication

Connect to your SSH server using GO-Global TN with the SSH protocol. When the connection succeeds you will be prompted for your user name and password to login. Once logged in, you must configure the server to accept your public key for authentication:

- If your server is using the SSH 1 protocol, you should change into the `.ssh` directory and open the file `authorized_keys` with your favourite editor. (You may have to create this file if this is the first key you have put in it). Then switch to the GO-Global TN Keygen window, select all of the text in the 'Public key for pasting into authorized_keys file' box (see section 6.2.10), and copy it to the clipboard (`Ctrl+C`). Then, switch back to the GO-Global TN window and insert the data into the open file, making sure it ends up all on one line. Save the file.

- If your server is OpenSSH and is using the SSH 2 protocol, you should follow the same instructions, except that in earlier versions of OpenSSH 2 the file might be called `authorized_keys2`. (In modern versions the same `authorized_keys` file is used for both SSH 1 and SSH 2 keys.)

- If your server is `ssh.com`'s SSH 2 product, you need to save a *public* key file from GO-Global TN Keygen (see section 6.2.9), and copy that into the `.ssh2` directory on the server. Then you should go into that `.ssh2` directory, and edit (or create) a file called `authorization`. In this file you should put a line like `Key mykey.pub`, with `mykey.pub` replaced by the name of your key file.

- For other SSH server software, you should refer to the manual for that server.

You may also need to ensure that your home directory, your `.ssh` directory, and any other files involved (such as `authorized_keys`, `authorized_keys2` or `authorization`) are not group-writable. You can typically do this by using a command such as

```
chmod g-w $HOME $HOME/.ssh $HOME/.ssh/authorized_keys
```

Your server should now be configured to accept authentication using your private key. Now you need to configure GO-Global TN to *attempt* authentication using your private key. You can do this in either of two ways:

- Select the private key in GO-Global TN's configuration. See section 5.18.5 for details.

- Load the private key into GO-Global TN Auth (see chapter 7). In this case GO-Global TN will automatically try to use it for authentication if it can.

Chapter 7: Using GO-Global TN Auth for authentication

GO-Global TN Auth is an SSH authentication agent. It holds your private keys in memory, already decoded, so that you can use them often without needing to type a passphrase.

7.1 Getting started with GO-Global TN Auth

Before you run GO-Global TN Auth, you need to have a private key. See chapter 6 to find out how to generate and use one.

When you run GO-Global TN Auth, it will put an icon of a computer wearing a hat into the System tray. It will then sit and do nothing, until you load a private key into it.

If you click the GO-Global TN Auth icon with the right mouse button, you will see a menu. Select ‘View Keys’ from this menu. The GO-Global TN Auth main window will appear. (You can also bring this window up by double-clicking on the GO-Global TN Auth icon.)

The GO-Global TN Auth window contains a list box. This shows the private keys GO-Global TN Auth is holding. When you start GO-Global TN Auth, it has no keys, so the list box will be empty. After you add one or more keys, they will show up in the list box.

To add a key to GO-Global TN Auth, press the ‘Add Key’ button. GO-Global TN Auth will bring up a file dialog, labelled ‘Select Private Key File’. Find your private key file in this dialog, and press ‘Open’.

GO-Global TN Auth will now load the private key. If the key is protected by a passphrase, GO-Global TN Auth will ask you to type the passphrase. When the key has been loaded, it will appear in the list in the GO-Global TN Auth window.

Now start GO-Global TN and open an SSH session to a site that accepts your key. GO-Global TN will notice that GO-Global TN Auth is running, retrieve the key automatically from GO-Global TN Auth, and use it to authenticate. You can now open as many GO-Global TN sessions as you like without having to type your passphrase again.

When you want to shut down GO-Global TN Auth, click the right button on the GO-Global TN Auth icon in the System tray, and select ‘Exit’ from the menu. Closing the GO-Global TN Auth main window does *not* shut down GO-Global TN Auth.

7.2 The GO-Global TN Auth main window

The GO-Global TN Auth main window appears when you left-click on the GO-Global TN Auth system tray icon, or alternatively right-click and select ‘View Keys’ from the menu. You can use it to keep track of what keys are currently loaded into GO-Global TN Auth, and to add new ones or remove the existing keys.

7.2.1 The key list box

The large list box in the GO-Global TN Auth main window lists the private keys that are currently loaded into GO-Global TN Auth. The list might look something like this:

```
ssh1     1024  22:c3:68:3b:09:41:36:c3:39:83:91:ae:71:b2:0f:04  key1
ssh-rsa  1023  74:63:08:82:95:75:e1:7c:33:31:bb:cb:00:c0:89:8b  key2
```

For each key, the list box will tell you:

- The type of the key. Currently, this can be `ssh1` (an RSA key for use with the SSH v1 protocol), `ssh-rsa` (an RSA key for use with the SSH v2 protocol), or `ssh-dss` (a DSA key for use with the SSH v2 protocol).
- The size (in bits) of the key.
- The fingerprint for the public key. This should be the same fingerprint given by GO-Global TNgen, and (hopefully) also the same fingerprint shown by remote utilities such as `ssh-keygen` when applied to your `authorized_keys` file.
- The comment attached to the key.

7.2.2 The ‘Add Key’ button

To add a key to GO-Global TN Auth by reading it out of a local disk file, press the ‘Add Key’ button in the GO-Global TN Auth main window, or alternatively right-click on the GO-Global TN Auth icon in the system tray and select ‘Add Key’ from there.

GO-Global TN Auth will bring up a file dialog, labelled ‘Select Private Key File’. Find your private key file in this dialog, and press ‘Open’. If you want to add more than one key at once, you can select multiple files using Shift-click (to select several adjacent files) or Ctrl-click (to select non-adjacent files).

GO-Global TN Auth will now load the private key(s). If a key is protected by a passphrase, GO-Global TN Auth will ask you to type the passphrase.

(This is not the only way to add a private key to GO-Global TN Auth. You can also add one from a remote system by using agent forwarding; see section 7.4 for details.)

7.2.3 The ‘Remove Key’ button

If you need to remove a key from GO-Global TN Auth, select that key in the list box, and press the ‘Remove Key’ button. GO-Global TN Auth will remove the key from its memory.

You can apply this to keys you added using the ‘Add Key’ button, or to keys you added remotely using agent forwarding (see section 7.4); it makes no difference.

7.3 The GO-Global TN Auth command line

GO-Global TN Auth can be made to do things automatically when it starts up, by specifying instructions on its command line. If you're starting GO-Global TN Auth from the Windows GUI, you can arrange this by editing the properties of the Windows shortcut that it was started from.

7.3.1 Making GO-Global TN Auth automatically load keys on startup

GO-Global TN Auth can automatically load one or more private keys when it starts up, if you provide them on the GO-Global TN Auth command line. Your command line might then look like:

```
C:\GO-Global TN\pageant.exe d:\main.key d:\secondary.key
```

If the keys are stored encrypted, GO-Global TN Auth will request the passphrases on startup.

7.3.2 Making GO-Global TN Auth run another program

You can arrange for GO-Global TN Auth to start another program once it has initialised itself and loaded any keys specified on its command line. This program (perhaps a GO-Global TN, or a WinCVS making use of Plink, or whatever) will then be able to use the keys GO-Global TN Auth has loaded.

You do this by specifying the `-c` option followed by the command, like this:

```
C:\GO-Global TN\pageant.exe d:\main.key -c C:\GO-Global TN\ggtn.exe
```

7.4 Using agent forwarding

Agent forwarding is a mechanism that allows applications on your SSH server machine to talk to the agent on your client machine.

Note that at present, agent forwarding in SSH2 is only available when your SSH server is OpenSSH. The `ssh.com` server uses a different agent protocol which they have not published. If you would like GO-Global TN to be able to support agent forwarding to an `ssh.com` server, please write to `ssh.com` and explain to them that they are hurting themselves and their users by keeping their protocol secret.

To enable agent forwarding, first start GO-Global TN Auth. Then set up a GO-Global TN SSH session in which 'Allow agent forwarding' is enabled (see section 5.18.3). Open the session as normal.

If this has worked, your applications on the server should now have access to a Unix domain socket which the SSH server will forward back to GO-Global TN, and GO-Global TN will forward on to the agent. To check that this has actually happened, you can try this command on Unix server machines:

```
unixbox:~$ echo $SSH_AUTH_SOCK
/tmp/ssh-XXNP18Jz/agent.28794
unixbox:~$
```

If the result line comes up blank, agent forwarding has not been enabled at all.

Now if you run `ssh` on the server and use it to connect through to another server that accepts one of the keys in GO-Global TN Auth, you should be able to log in without a password:

```
unixbox:~$ ssh -v otherunixbox
[...]
debug: next auth method to try is publickey
debug: userauth_pubkey_agent: trying agent key my-ssh-key
```

```
debug: ssh-userauth2 successful: method publickey
[...]
```

If you enable agent forwarding on *that* SSH connection as well (see the manual for your server-side SSH client to find out how to do this), your authentication keys will still be available on the next machine you connect to - two SSH connections away from where they're actually stored.

In addition, if you have a private key on one of the SSH servers, you can send it all the way back to GO-Global TN Auth using the local `ssh-add` command:

```
unixbox:~$ ssh-add ~/.ssh/id_rsa
Need passphrase for /home/fred/.ssh/id_rsa
Enter passphrase for /home/fred/.ssh/id_rsa:
Identity added: /home/fred/.ssh/id_rsa (/home/me/.ssh/id_rsa)
unixbox:~$
```

and then it's available to every machine that has agent forwarding available (not just the ones downstream of the place you added it).

7.5 Security considerations

Using GO-Global TN Auth for public-key authentication gives you the convenience of being able to open multiple SSH sessions without having to type a passphrase every time, but also gives you the security benefit of never storing a decrypted private key on disk. Many people feel this is a good compromise between security and convenience.

It *is* a compromise, however. Holding your decrypted private keys in GO-Global TN Auth is better than storing them in easy-to-find disk files, but still less secure than not storing them anywhere at all. This is for two reasons:

- Windows unfortunately provides no way to protect pieces of memory from being written to the system swap file. So if GO-Global TN Auth is holding your private keys for a long period of time, it's possible that decrypted private key data may be written to the system swap file, and an attacker who gained access to your hard disk later on might be able to recover that data. (However, if you stored an unencrypted key in a disk file they would *certainly* be able to recover it.)
- Although, like most modern operating systems, Windows prevents programs from accidentally accessing one another's memory space, it does allow programs to access one another's memory space deliberately, for special purposes such as debugging. This means that if you allow a virus, trojan, or other malicious program on to your Windows system while GO-Global TN Auth is running, it could access the memory of the GO-Global TN Auth process, extract your decrypted authentication keys, and send them back to its master.

Similarly, use of agent *forwarding* is a security improvement on other methods of one-touch authentication, but not perfect. Holding your keys in GO-Global TN Auth on your Windows box has a security advantage over holding them on the remote server machine itself (either in an agent or just unencrypted on disk), because if the server machine ever sees your unencrypted private key then the sysadmin or anyone who cracks the machine can steal the keys and pretend to be you for as long as they want.

However, the sysadmin of the server machine can always pretend to be you *on that machine*. So if you forward your agent to a server machine, then the sysadmin of that machine can access the forwarded agent connection and request signatures from your public keys, and can therefore log in to other machines as you. They can only do this to a limited extent - when the agent forwarding disappears they lose the ability - but using GO-Global TN Auth doesn't actually *prevent* the sysadmin (or hackers) on the server from doing this.

Therefore, if you don't trust the sysadmin of a server machine, you should *never* use agent forwarding to that machine. (Of course you also shouldn't store private keys on that machine, type passphrases into it, or log into other machines from it in any way at all; GO-Global TN Auth is hardly unique in this respect.)

Appendix A: GO-Global TN Licence

GO-Global TN is copyright 2001-2002 by GraphOn Corporation.

Portions copyright Simon Tatham, Robert de Bath, Joris van Rantwijk, Delian Delchev, Andreas Schultz, Jeroen Massar, Wez Furlong, Nicolas Barry, and CORE SDI S.A.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Appendix B: GO-Global TN QuickStart

This is the GO-Global TN QuickStart document, designed to help the first-time user quickly and easily install and run the GO-Global TN product. For more complete instructions, consult the *GO-Global TN User Manual*.

B.1 System Requirements

To install and run the GO-Global TN server software, your UNIX host must have at least the following resources available:

- Up to 5MB disk space
- 10MB of temporary disk space in /tmp

B.2 Installing GO-Global TN

The following steps will install the GO-Global TN software on your UNIX host and on your Windows PC. These instructions assume some familiarity with the UNIX operating system.

B.3 Installing the Host Software

To install the UNIX host portion of GO-Global TN, follow these steps:

1. Become root.
2. Locate the GO-Global TN host installer package, usually named GGTN_*platform*.bin
3. If you transferred the host installer package, make sure it is executable.
4. Execute the binary.
5. Follow the instructions given by the host installer.

You have completed the server installation.

B.4 Installing the Client Software

To install the Windows client portion of GO-Global TN, follow these steps:

1. Locate the GO-Global TN client installer package, usually named GO-Global TN.exe.
2. Execute the installer by double-clicking on it.
3. Follow the instructions given by the client installer.

You have completed the client installation.

B.5 Running GO-Global TN

To run the GO-Global TN software, run the GO-Global software. It is located in the *Start* menu, under Programs, GraphOn, GO-Global TN, GO-Global TN.

Use the client software to connect to your UNIX host, over Telnet or SSH.

Once logged into the UNIX host, run ‘ggtn’ from the shell command prompt.

GO-Global TN will then begin running and you will see several messages like:

```
-- GO-Global TN Telnet Tunnel --
Version 2.1.0 beta

Copyright (C) 1999-2001 GraphOn Corporation
All Rights Reserved.

$
```

At the same time, the tunnels specified in your `ggtnrc` file are established, and any `Launch` commands present are executed. The GO-Global TN window will launch a sub-shell that can be used to issue commands to the remote UNIX host while the tunnels are in use.

If you are using the default installation of GO-Global UX and GO-Global TN, you should see GO-Global UX begin running and ask you to log into the remote host. Enter your user name and password to continue.

From this point forward, you should use GO-Global UX as though you were directly connected to the remote UNIX machine.

Note that if you exit the GO-Global TN window (e.g. by typing ‘exit’) then the host portion of GO-Global UX will shut down and your tunnels will close, causing GO-Global UX to stop or suspend its currently running sessions, if any. You should not exit the initial GO-Global TN SSH or Telnet window until you are ready to close all of your tunneled connections.