· Guide to Using

SecureNetTerm

By InterSoft International, Inc.

Contents

Global Settings 27

Site Profile Manager 35

Advanced Host Settings 37

ActiveX Scripting 47

SecureNetTerm ActiveX Object 51

Introduction

SecureNetTerm

SecureNetTerm is an interactive communications program designed for communication with hosts supporting the telnet, SSH and SSL/TSL network protocols. SecureNetTerm supports all the popular terminal emulations used today.

A common question is what emulation do I need? The answer to that depends upon the host that you connect to. Most sites will document the required emulation in their access their system, while others will provide a menu selection at connect time, which allows you to specify the emulation type. In either case you must provide the emulation type in the site profile for that host. A good selection is ANSI if you are not sure.

The Site Profile Manager maintains the site profiles, which contain all the necessary information required to access a host, such as userid, password, protocol, host name, port, etc. The host name consists of a fully qualified network name or an IP address, which is used to identify the host. Each computer on a network, connected directly or through the Internet, has a unique name and IP number. Think of the network name as you would your name. Then think of the IP address as your telephone number. As with the telephone system, you cannot connect by name but by phone number. However if SecureNetTerm knows either one, it can determine the other using a technique similar to looking it up in the white pages. The only thing important is that one must be entered into the site profile. Using the network name is of course the best way, since if the IP number changes, SecureNetTerm can still connect by looking up the new number based upon the qualified network name. IP numbers are just as dynamic as phone numbers, but the network name normally remains the same.

In addition, each site profile contains all the related security information, such as authentication type, public/private keys and/or certificates to use to automate the login process. The Security Manager is used to define and change security related information.

Hosts that are only accessed occasionally can be connected to using the “QuickConnect” dialog bar.

Whenever SecureNetTerm is connected to a host, the animation at the upper right of the screen will be active. This acts as a visual reminder to the user that SecureNetTerm has an active, open connection to a host. The animation can be turned off within the View menu item.

Mouse Panning

SecureNetTerm supports mouse panning for both horizontal and vertical scrolling. Mouse panning is enabled by pressing the middle mouse wheel or by selecting the Mouse Panning menu item from the Options menu. The speed of the pan is controlled by the distance of the cursor from where you started panning.

Dynamic Views

SecureNetTerm has the ability to display one host per window, or multiple hosts per window. Displaying multiple hosts within a single window is referred to as Dynamic Tab Views, and can be enabled with the View-Dynamic Tab Views menu item. The View-Window Tab Bar menu item controls the display of the window tab bar which allows for selection of the active window.

If the Dynamic Tab Views option is not selected, a single host will be displayed in the SecureNetTerm window. Additional hosts can be connected using multiple windows. The Windows Tab Bar can be used to switch between the multiple windows, as well as the Window menu. In addition, the Window menu contains the options "New Window", "Tile Windows" and "Cascade Windows" to manange the multiple windows.

If the Dynamic Tab Views options is selected, multiple hosts will be displayed in a single SecureNetTerm window. The Windows Tab Bar is used to switch between the multiple hosts. The ability still exists to have multiple SecureNetTerm windows, each containing a singe host or multiple hosts.

Note that the Windows Tab Bar must be enabled prior to enabling the Dynamic Tab Views option.

Font Facts

The use of fonts within a terminal emulation program tends to generate a lot of confusion and requests for assistance. This short discussion of fonts should help you understand the importance of fonts, how they affect a terminal emulation program, and what impact they have upon what gets displayed on the screen.

First, emulation programs use fixed pitch fonts, just like the real terminals they emulate. This is in contrast to most Windows based editors that use proportional fonts. A fixed pitch font simply means that all the characters that get displayed have the same width and height, resulting in a display that always has the same width and height for a given number of rows and columns. The number of fixed pitched fonts is few, when compared to the number of proportional fonts. The most common are the ones supplied with this program (NetTerm Ansi and NetTerm OEM), and the Microsoft fonts Courier New, Lucida Console and Terminal.

Fixed pitched fonts also implies that the ratio of width verses height be maintained at a fixed ratio. When the terminal emulator program window is dragged by the user, the number of rows and columns are normally held constant. The font size is increased/decreased, while maintaining the fixed ratio. When the dragging action has been completed, the program computes the closest font size, then computes the width of a frame to enclose the resulting display. The role of the frame is to maintain the proper font ratio for the specified window size.

The window size is also affected whenever the user, or host, changes the number of rows or columns. In this case, the font is held constant and the number of rows/columns is changed.

SecureNetTerm supports two major font/window size models. That is, you can select to change the font size, or the number of rows/columns when the window size is changed. The most common method is to keep the rows/columns constant, and change the font size.

Fonts are used to translate a data character sent from the host to a visible item on the screen. Internally, the emulator program is not concerned with what is being received, or what is being displayed. It simply receives the data, and instructs the system to display the character associated with its value from the font that is currently selected. Most emulator programs, including this one, only support Ansi and extended Ansi characters sets. This implies it supports the range of characters with a numerical value of 0 to 127 for Ansi and 0 to 255 for extended Ansi. An important thing to remember is just because the emulator program supports the character range of 0 to 255 does not imply that the host or the host application programs supports that range.

Historically, hosts and the terminals connected to them, used the Ansi character set, which restricted the range of characters from 0 to 127. This range was not sufficient to handle language unique characters, which resulted in special character translation tables on the host and within terminals. The most common is the set of tables referred to as the National Replacement Characters (NRC). SecureNetTerm supports the National Replacement Characters, as well as the ability to define user specific keyboard and video lookup tables.

Text Selection

SecureNetTerm has three modes of text selection, consisting of word highlighting, current page text selection and multi-page text selection. A word is defined as a series of data characters without embedded blanks.

A word can be selected by double clicking the word with the left mouse button. The most common use of this is to select a word, then use the right mouse menu Copy/Paste option to copy the word to the clipboard, then paste it at the current cursor location in one operation.

Current page text selection allows for the selection of data within the current page (that data that can be seen on the screen) by pressing the left mouse button and then moving the mouse to highlight the desired text. Note that this mode uses a modified block mode.

Multiple page text selection uses the block mode concept. The text selection is initiated by a shift-left mouse click at the upper left corner of the text to be selected. A shift-left mouse click at the lower right corner completes the text selection. The scroll bar is used to scroll the data between the initial and final shift-left mouse clicks. This mode can be used on the current page as well as multiple pages of data.

Help File Text Size

The text size within the right pane of this Microsoft help file can be changed with the Microsoft Browser View-Text Size menu item. If you change the text size with the Browser, you must close this help file, and display it again to reflect the new text size.

System Requirements

A minimum of a 486-based machine with 16 megabytes of RAM. The speed of the machine, and the available RAM will have a direct relation to file transfer rates.

Windows® 2000, 2003, 2008, XP®, Vista®, Windows 7, Windows 8, and Windows 10.

Winsock.dll or wsock32.dll compliant with version 1.1 or above.

A minimum screen resolution of 800x600 pixels.

A video card with a 16 bit (medium) or 24 bit (high) color quality support. Video cards with less than 16 bit color quality will result in poor icon resolution.

Control Information

SecureNetTerm maintains three user specific files, SecureCommon.xml, SecureCommon.ini and the known_hosts file. These three files are placed in the users specific application data directory during the program installation by the Installshield installation program, when the Installshield program is run by the end user. If the program is installed by the System Administrator, refer to the System Administration Issues section.

The location of the application data directory differs with the various types of Microsoft systems; common locations are:

Windows XP	C:\Documents and Settings\<user>\Application Data\InterSoft Common
Windows Vista	C:\Users\<user>\AppData\Roaming\InterSoft Common
Windows 7	C:\Users\<user>\AppData\Roaming\InterSoft Common
Windows 8	C:\Users\<user>\AppData\Roaming\InterSoft Common
Windows 10	C:\Users\<user>\AppData\Roaming\InterSoft Common

The term <user> is unique for each individual that has been assigned a userid on the workstation.

The SecureCommon.xml file contains all the information that is unique for each host and associated security information. The SecureCommon.ini file contains global information that is common to all hosts. The known_hosts file contains a unique identifier for each host connected to with the SSH protocol. If you have a need to migrate to another workstation, or to install SecureNetTerm on another system and desire to maintain this information, simply copy the three files to the new system after you install SecureNetTerm.

As with any other mission critical data file, it is wise to create backup copies of these files on a regular basis.

Items that are of a temporary nature, or that can be easily reset, are maintained in the system registry under the key:

HKEY-CURRENT-USER/Software/InterSoft International, Inc./SecureNetTerm

This includes such items as the window size, window position, state of ToolBar icons, size of each unique display area and other display characteristics.

The Installshield installation routine will determine the proper location of the SecureNetTerm control files and will insert their locations in the following registry keys:

HKEY-CURRENT-USER-Software/InterSoft International, Inc./SecureNetTerm/UserIniFile
HKEY-CURRENT-USER-Software/InterSoft International, Inc./SecureNetTerm/UserSiteFile
HKEY-CURRENT-USER-Software/InterSoft International, Inc./SecureNetTerm/UserKnownHosts

Changing these keys can change the location of the SecureCommon.ini and SecureCommon.xml files.

Internationalized Domain Names (IDN)

SecureNetTerm supports IDNA (Internationalizing Domain Names in Applications) for domains with unicode characters. IDNA is a mechanism to represent non-ASCII domains with only ASCII characters. This feature is transparently implemented, domain names with unicode characters are automatically converted to the IDN format.

For Windows 95/98, install Microsoft's Windows Installer Redistributable which will update your system's RichEdit Control to version 3.0. If you need support for multibyte languages, such as Chinese and Korean, you will need a unicode truetype font such as "Arial Unicode MS" from Microsoft.

References
VeriSign IDN-client for IE
IDN Convertor Test

RFC
String Preparation (stringprep)
IDNs in Applications (IDNA)
Name Preparation (nameprep)
Encoding Scheme (punycode)

Library and SDK
VeriSign IDN Software Development Kit

Internationalized Keyboard Configuration

Users that have an XP system, that have a need to enter international characters can use the following tip:

1. Start -> Settings -> Control Panel
2. Regional and Language Options
3. Languages -> Details ...
4. Click Add.
5. Under Input language, choose your native language.
6. Under Keyboard layout/IME, choose United States-International.

Now to form accents, you prefix the letter with either **` ' " or** ^ So, to get è, one types ` and then e. To get Ë, one types " and then E.

System Administration Issues

The Installshield installation routine normally determines the proper location of the SecureNetTerm control files (see Control Information) and will insert their locations in the following registry keys:

HKEY-CURRENT-USER-Software/InterSoft International, Inc./SecureNetTerm/UserIniFile
HKEY-CURRENT-USER-Software/InterSoft International, Inc./SecureNetTerm/UserSiteFile
HKEY-CURRENT-USER-Software/InterSoft International, Inc./SecureNetTerm/UserKnownHosts

In addition, the three control files (as distributed from InterSoft) will be copied from the program installation directory to the locations pointed to by these registry entries

If it is desired to distribute custom versions of the three control files, they should be placed in the installations directory (replacing those placed there by Installshield on the initial installation by the system administrator). Subsequent installations performed by Installshield will then use the custom files. System administrators should consult Installshield and Microsoft documentation for detailed instructions on unique installation requirements.

If the program cannot locate the user specific SecureCommon.ini, SecureCommon.xml and known_hosts files upon initial startup, the users unique application directory will be determined with the Microsoft API ShGetSpecialFolderPath using a nFolder value of CSIDL_APPDATA. The program will then copy the SecureCommon.ini, SecureCommon.xml files and the known_hosts files located in the program installed directory to the users unique application directory. The registry will then be updated to reflect the correct location of these files.

Copyright and Trademarks

SecureNetTerm™ Copyright © 1995-2016 InterSoft International, Inc. All Rights Reserved

SecureNetTerm™ is a registered trademark of InterSoft International, Inc. in the United States and/or other countries.

SSH® and SSH2™ are registered trademarks or trademarks of SSH Communications Security Oyj in the United States and/or other countries.

Windows® and Windows NT® are registered trademarks of Microsoft Corporation in the United States and/or other countries.

How To

Show a Logo

SecureNetTerm can display a logo whenever a connection is not active. The logo file must be in a BMP (bitmap), JPEG, or GIF format. Control of the logo display, and the file containing the image, is defined in the Global Settings-Logo dialog which can be accessed from the toolbar or the Options menu.

It the image is a GIF format, you have the option of designating one color in the image as transparent when that image is created. This transparency feature is normally used to make it appear as if an irregularly shaped image is floating on the background. That is, if the image selected for display is a GIF format, with a transparent color, the transparent color will be changed to the current SecureNetTerm background color whenever it is displayed. The transparent GIF feature is only supported by Microsoft on Windows XP based systems.

The transparency feature is also supported with a BMP (bitmap) file on all versions of Windows. The bitmap file should have a single color reserved as a background or transparent color. In the Logo dialog, select the "Enable Image Mask Color" option, and then choose the image background or transparent color. When the image is displayed, this color will be changed to the current SecureNetTerm background color.

Create a new profile

SecureNetTerm is normally distributed with example configurations for many of the host sessions that it supports. Each protocol, such as telnet, SSH-1, SSH-2 and TLS/SSL has many configurable options that must be correct for a successful connection. The example host profiles provide a template which can be used to create a new host profile with a minimum of effort.

To add a new host profile, press the “Site Profile Manager” icon on the ToolBar. This will open up the Site Profile Manager. The Site Profile Manager maintains folders and profiles within those folders for use with both SecureNetTerm and SecureFTP. Profiles that are used for both are referred to as dual use.

On the left side of the window is a tree containing folders associated with a protocol, and example host entries for that protocol. The Interactive folder contains examples for the telnet protocol, direct connect and dialup hosts. The MyFTPS folder contains examples for TLS/SSL hosts. The mySFTP folder contains examples for the SSH protocol, including SSH-1 and SSH-2, for connecting to SSH server software from OpenSSH and SSH Data Communications.

To add a new site, select an current folder (or add a new folder) and press the "New Site" button. If you first highlight a current site, then press the shift key at the same time you press the "New Site" button, the highlighted site will be duplicated. Then provide the new site with a descriptive name. The next step is to provide all the required connection information for the new site.

The shift-New Site is one of the most powerful features of the Site Profile Manager. It allows you to create a new host profile, exactly like the one you currently have, by just changing the host name after the new profile has been created. In most cases this is all that is needed for those that always connect to hosts with the same Operating System, using the same protocol, such as SSH.

Userid/Password Considerations

Every host protocol supported by SecureNetTerm requires a valid userid for connecting to a host. Most protocols require a password. The fact that these two items are user dependent requires special consideration for both SecureNetTerm and the user. Each protocol has different methods for providing these two items, which a vastly different.

The telnet protocol, as well as the direct connect and dialup hosts ask for the userid and password interactively. The format, method and timing of how each host type requests this information is not consistent, requiring each to be supplied manually by the user, or by scripts created by the user.

For more advanced protocols, such as SSH, both the userid and password are passed to the host in a well defined method. The remainder of this topic pertains to the SSH protocol.

If the userid and or password is the same for most, if not all, of the hosts you connect to, then they should be specified in the Global Settings-Connection-Control panel. If you have a few hosts that have a different userid and or password, you can place them in the host profile for those few hosts, or you can have SecureNetTerm request them at connect time. Whenever the host requests either the userid or password, SecureNetTerm will first check the host profile, then the global definition. If they are not contained in either location, each will be requested interactively.

Connect to a host using a profile

SecureNetTerm has the ability to remember the last profile you connected to, as well as a default host. When you start SecureNetTerm, and you want to connect to the last active host, simply press the “Connect” icon on the ToolBar. To select a new host profile, press the “Site Profile Manager” icon, and then select the desired host. You can then press the “Connect” button in the Site Profile Manager, or you can exit the Site Profile Manager, then press the ToolBar “Connect” icon.

If you select the "Default Host" checkbox for a host, it will always be displayed when you open the Site Profile Manager.

Connect to a host without a profile

You can connect to any host without the need to create a host profile using the QuickConnect bar. This is useful for those hosts that are rarely accessed.

The QuickConnect bar can be used to connect to all types of host servers. It also allows for userid/password authentication for each server type. If a SSL/TLS server requires a client certificate, a popup dialog will be presented when the server requests the client certificate. The popup dialog will display a list of your certificates, and allows you to view/select the proper one to use.

To enter a new host, simply type over the current host name and the other fields. When you press the "QuickConnect" button (the first button on the left side of the bar) to connect, the new host will be entered into the QuickConnect history list. This list can be cleared at any time by pressing the "QuickConnect" button while holding down the Shift-Delete keys. The current entry can be deleted by pressing the "QuickConnect" button while holding down the Ctrl-Delete keys.

Use X.509 Certificates

A X.509 digital certificate is a set of electronic credentials that uniquely identify an individual. There are two parts to a digital certificate: a private key and a certificate.

Your private key is the piece of information that uniquely identifies you within the Public Key Infrastructure. The private key is mathematically created on your personal workstation, under your supervision, and is known only to you.

The certificate is the public part of your digital certificate. It contains your name and other identifying information. It also contains the public key, which is mathematically related to the private key. Using your certificate, other people can verify that you hold your private key, and therefore, must really be who you say you are.

Certificates can be generated and stored within browser software stores (databases), or within specialized Smart Cards/USB tokens. Smart Cards/USB tokens are hardware devices that protect the private key from access by anyone accept the owner. This is the main benefit of these devices. They serve as an impenetrable safe for the private key, ensuring that only the intended user has access to it. The private key can be generated on-board and never leaves the device for signing and encryption operations.

Certificates can be used for authentication with servers supporting SSH and SSL/TLS.

Manage SSH Host Keys

The SSH authentication allows public/private RSA keys, as well as X.509 certificates to be used for client authentication. Although the documentation of many host SSH servers only refer to public/private RSA keys, X.509 certificates can still be used, since these certificates are nothing more than public/private keys with enhanced security related information contained within a “certificate”. The public key is a component of this certificate. SecureNetTerm fully supports the use of certificates for public/private key authentication.

The concept of public/private key authentication is a public/private key pair is created on the workstation. The private key is retained on your local workstation, the public key is transferred to the host and placed in a unique location specified by the host SSH server. When you attempt to login to a host using public/private key authentication, the host server will send a challenge to SecureNetTerm. SecureNetTerm will sign this challenge with your respective private key and return the signed challenge to the host server. The server will then verify the signed challenge using your public key. If the host server determines that you are the holder of the correct private key, it will allow you to login. Refer to the Key Management section on how to create and install these key pairs.

Manage SSL/TLS .tlslogin file

Most UNIX based SSL/TLS servers allow logging into the system with only a userid and a certificate. This is commonly referred to as client certificate authentication. Although the complete details/requirements on how to enable this feature can only be provided by the system administrator of the host, almost all of these systems require a copy of the client certificate be located on the host. The .tlslogin file, located in the users home directory, is commonly used to contain these user certificates. Most servers allow multiple certificates to be placed within the file.

Once you have obtained a user certificate for your workstation, and have selected that certificate to be used for SSL/TLS access to a host, you can export that certificate to a file, then copy that file to the host and place it in the .tlslogin file. Refer to the SSL/TLS security section on how to define, create and install the certificate.

Program Overview

Menus

File Menu

The file menu supports connecting/disconnecting a host, access to the Site Profile Manager, printer management and program exit. The "Process SmartPrint" item is normally grayed out indicating that no printer output is pending within SmartPrint. This item will be enabled whenever the SmartPrint option is enabled, and printer output has been received form the host. If enabled, this option should be selected prior to closing SecureNetTerm to send the contents of the SmartPrint buffers to a printer.

Edit Menu

The edit menu provides copy/paste clipboard support and special terminal requirements.

The copy/paste menu items support both the active (current) screen as well as the scroll back buffer. The scroll back buffer contains data that was once displayed on the active screen, and was then "scrolled" out of view by the contents of a new screen. In the past, the scroll back buffer provided an exact duplicate of lines that were displayed on the screen. This was in the era where the host wrote physical lines to the screen (a physical line was one that contained line termination characters such as an line feed and or a carriage return). Most current applications use the concept of full screen mode in which data is placed upon the screen based upon the current cursor location. Data written to the screen in this manner cannot be placed within the scroll back buffer.

The special terminal menu item "Reset Cursor" will restore (display) the terminal cursor if it is hidden for some reason. The host application has the ability to hide the cursor, and sometimes forgets to turn it back on. Under most conditions, this menu item will not have to be used.

The special terminal menu item "Reset Terminal" is much like the previous item, in that it can be used when a host application is aborted and leaves the terminal in an unknown state.

The Send Short and Long Break menu items are used to send a break signal. If the current connection is over a modem, a modem break hardware signal will be sent. If the current connect is over a telnet channel, a telnet break command will be sent.

The "Telnet Abort" menu items sends a telnet abort command, which will stop the current running host application. This command is only useful for telnet connections.

View Menu

The view menu allows the Toolbar, animation, StatusBar, Terminal Status line and QuickConnect items to be toggled on/off. The Office XP Look menu items controls the look of items such as the Toolbar.

The Binary mode item toggles the active SecureNetTerm session between normal 7-bit ASCII mode and 8-bit binary mode.

The Session Logging item toggles session logging on/off. See session logging for additional details.

Tools Menu

The tools menu provides the ability to define commonly used programs, which can be started from within SecureNetTerm. In addition, the customize menu item provides access to a User Management panel, which allows for user customization of SecureNetTerm. The User Management panel contains four tabs; Commands, Keyboard, Tools, and Options.

The tools tab allows you to define commonly used programs, which you can start from within SecureNetTerm. Simply select the customize menu, select the tools tab, then press the New (Insert) icon. Then add a description name for the program you wish to add to the menu, and then define the directory/path to that program. The descriptive name you selected for the program will appear as an entry within the tools menu.

The Keyboard tab provides an easy way for you to associate an accelerator key to a menu or toolbar item.

Options Menu

The Options menu, consisting of five groups, is designed to provide easy access to SecureNetTerm utilities and program support.

The first group, certificate support, provides limited X509 certificate support for those that use the SSL/TLS interactive host servers. The first item, Certificates…, will display all the certificates owned by the current computer user which are contained within the Microsoft Certificate Store. Those companies that allow user created access certificates to be used as valid certificates for host authentication may use the second menu item, Create User Certificate. The third menu item, Show Server Certificate… will display the certificate presented by the host computer at login time.

The second group, SSH utilities, are active whenever a connection is made to a SSH style server.

The next three groups are used to start commonly used programs and to customize portions of SecureNetTerm.

Script Menu

The script menu controls running scripts on demand. The Run menu item will bring up a file open dialog to select the desired script. Once the script is opened, it will be passed to the script engine for processing. The Stop menu item will stop the current running script.

The bottom half of the menu contains a "MRU" (most recently used) selection list of the last ten scripts run. These scripts can be run by selecting the desired script from the list.

The MRU list can be cleared by pressing the shift key concurrently with the Run menu item.

Language Menu

The language menu allows for the selection of the GUI desired language.

Window Menu

The window menu keeps track of multiple instances of SecureNetTerm, and allows you to switch to another instance. The “Save Window Position” menu item will save the current screen location of the window. Subsequent launching of this host profile will place the window in the saved position. The saved positon can be cleared in the Advanced Host Settings-Extended Options dialog.

The "Tile Windows" menu item will place all windows on top of each other, starting with the x,y location of the first active SecureNetTerm window. This is useful for multiple SecureNetTerm windows being placed in the same location on the screen, and using the Window Tab Bar to toggle between the windows.

The "Cascade Windows" menu item will cascade all SecureNetTerm windows starting with the active windows upper left top position. The active window will remain the top most window of the cascade.

Help Menu

The help menu contains support for displaying this help file, displaying/enabling the tip file, registering the program and www browser access to SecureNetTerm's home page. The about menu item displays the current program version information.

ToolBar

The toolbar, located at the top of the main window, provides one-click access to commonly used functions. The toolbar contains seventeen icons, divided into five major groups.

Connect	Connect to the active host.
Host History	Maintains a MRU of connected hosts. (See below)
Disconnect	Close the current connection.
Site Profile Manager	Start the Site Profile Manager
Advanced Host Settings	Bring up the Advanced Host Setting dialog
Global Settings	Start the Global Settings Dialog
Keyboard Keys	Start the Keyboard Key definition dialog
Font	Change the active host font
Print Screen	If connected, print the current screen
Save Screen	If connected, save the current screen to a file
Copy	If connected, save the current screen to the clipboard
Paste	If connected, paste the clipboard to screen
Script Editor	Start the internal script editor
Run Script	Run the current script
Stop Script	Stop the current script
Start SecureFTP	Run the SecureFTP program. (See below)
Abort Transfer/Connect	Abort Request (See below)

The Host History dropdown maintains a MRU (Most Recently Used) list of hosts, which have been connected to. You can connect to the hosts within this list by simply selecting a host from the list. If SecureNetTerm is already connected to a host, a new instance of SecureNetTerm will be used to connect to the selected host. Pressing the Host History button, when connected, will start a new instance of SecureNetTerm to the currently connected host. It the shift key is pressed when the Connect Dropdown is selected, the MRU list will be cleared.

The Script Editor, Run Script and Stop Script items allows for the loading, editing and running of a script on a one time basis. Once a script has been selected, it will be "remembered", by host.

The Start SecureFTP item will start the SecureFTP program. The installation location of SecureFTP must be defined in the Global Settings-Applications dialog. If SecureNetTerm is not connected to a host, SecureFTP will be started normally, that is, without command line input. If SecureNetTerm is connected to a host, SecureFTP will be started and passed the active profile. SecureFTP will then begin a connection to the host specified by that profile. For example, if SecureNetTerm is connected to a SSH based host with an active profile name of \mySFTP\SSH-2, then SecureFTP will be started with a command line input of –profile \mySFTP\SSH-2, which will instruct SecureFTP to connect to the SSH host specified by the profile. See the section on Dual Use hosts in the Site Profile Manager.

The Abort Transfer/Connect icon has several different functions, depending upon the state of SecureNetTerm. If a connection is being attempted, pressing this icon will abort the connection attempt and all connection retries. If a file transfer is in progress, it will attempt to abort the transfer. If SecureNetTerm is displaying a large amount of data being sent from the host, you can use the Abort icon to stop the flow of the data, and request the host to stop the application that is generating the output. When used for this purpose, wait a few seconds after you press the Abort button. This will allow time for the host and network to clear its buffers. Then press the Abort button again to resume normal operations.

The ToolBar can be turned off in the View menu.

StatusBar

The status bar is composed of nine informational areas consisting of:

1.	General Messages
2.	File Transfer progress meter
3.	Emulation Type
4.	Security indicator
5.	Current time or Row/Column
6.	LED display
7.	Printer Active/Keyboard locked indicators
8.	Caps Lock indicator
9.	Num Lock indicator

The StatusBar can be turned off in the View menu.

Right Mouse Menu

The right mouse menu contains unique and selected options within SecureNetTerm that are commonly used, and places them in one easy to access location. The menu consists of:

Copy	Copy highlighted text to the clipboard.
Paste	Paste contents of clipboard to current screen.
Copy/Paste	Perform copy/paste in one operation.
Launch Highlighted URL	Proccess the highlighted URL
Copy Screen	Copy current screen to clipboard.
Edit Screen	Edit the current screen with the user defined editor.
Save Screen	Save current screen to a file.
Clear Screen	Clear the current screen.
Print Screen	Print the current screen.

Copy Scroll Buffer	Copy the scroll buffer to the clipboard.
Edit Scroll Buffer	Edit the scroll buffer with the user-defined editor.
Save Scroll Buffer	Save the scroll buffer to a file.
Clear Scroll Buffer	Clear the current screen and the scroll buffer.
Print Scroll Buffer	Print the scroll buffer.
Edit Highlighted Text	Edit the highlighted text with the user-defined editor.
Print Highlighted Text	Print the highlighted text.

Command Line

When started, SecureNetTerm will check for command line arguments. Valid arguments are an optional profile name, an alternate .xml file (the Site Profile File) and/or a valid URL to use. If specified, the profile name must be specified with:

```
-profile
```

and it must specify a unique profile name to use for an immediate connection.. If an alternate Site Profile File is desired, it must be preceded with:

```
–i
```

The following are valid combinations:

```
-profile \Telnet\MyHost
-profile "\Telnet\Telnet Test"
-i \c:\temp\SecureCommon.xml
```

If the profile or Site Profile File name contains spaces, the name must be enclosed within quotes.

Note that the profile name can be obtained from the Site Profile Manager, "Profile Name" text box.

SecureNetTerm can also be passed a telnet style URL on the command line. If this option is selected, it must be the only item on the command line. A telnet URL follows the common Internet scheme syntax, defined as:

```
URL-type://<userid>:<password>@<host>:<port>/<url-path>?<typecode>
```

The URL-type must be telnet.

If the optional port is omitted, the port defaults to 23. If a port of 22 is specified, the protocol will be set to SSH-2.

If either the optional userid/password is omitted, and it is required by the host during the connection phase, it will be requested by SecureNetTerm.

The optional ?<typecode> specifies the data transfer format, and <typecode> can be one of the characters "i" or "a", representing binary or ASCII data respectively. These are not supported at the current time.

An example telnet URL would be:

```
telnet://zkrr01:mypass@secure.netterm.com:22
```

The optional <url-path> can be used to specify a default host profile to use for the connection. The character '/' is not a part of the <url-path>, it is a required deliminer if the <url-path> is specified. If the optional <url-path> is specified, the optional userid, password, host and port will override the values in the profile specified by the <url-path>. An example URL containing a default profile would be:

```
telnet://zkrr01:mypass@secure.netterm.com:22/\mySSH\OpenSSH-Password
```

Host Mouse Support

SecureNetTerm supports two models for host mouse support. The first is the standard XTERM model and is active only when the XTERM emulation is selected. The second is the Locator Input Model for ANSI Terminals, which is for all other emulations. In both models, the host must activate the mouse support. If the host has not activated mouse support, then normal Windows style mouse support will be in effect. Note that pressing the right mouse button will activate the right mouse popup menu containing many of the same items contained within the SecureNetTerm menu. The UNIX based program "Midnight Commander" (mc) is a good example of a host application, which supports XTERM style mouse support.

Host Editing

The host-editing feature allows text-based files to be downloaded from the host, and passed to a user-defined editor for modification or viewing. This feature requires a special program (named netedit.c) to be uploaded to your host and then compiled and linked. Simply transfer the file netedit.c (located in the SecureNetTerm directory) to your host then issue the following commands:

```
cc netedit.c -o netedit
chmod +x netedit
```

On some systems, you can also replace the first line with **make netedit**, then do the **chmod +x netedit** command.

To edit a host file, simply enter 'netedit xxxxxx' where xxxxxx is the file name to be edited. The file will then be transferred to your system, and the user-defined editor will be started with the file name received as a command line input. The Windows program 'Notepad' is sufficient for most cases. If you make any changes to the file, be sure to save it. SecureNetTerm can then detect that changes were made, and will then upload the file back to the host system. If no changes are made, SecureNetTerm will inform the host base netedit program of that fact. If changes are made, the host-based netedit program will create a new file, and if successful, it will replace the current file with the new file.

The host-editing feature described above uses the same logic as transparent printing to transfer files to the local machine and uses the same basic logic as ASCII file transfers to return the file back to the host. Although effective on most UNIX machines, some text files contain data that will prevent successful uploads back to the host. Text files containing 'long lines' such as those produced by word processors or some HTML editors cause this.

A second, more powerful way for editing host files is with the SmartEdit feature. This method uses the standard Zmodem file transfer to move files between the host and SecureNetTerm for editing, thus allowing binary files such as bitmaps and other graphic files to be edited with a graphical editor. Since the standard Zmodem file transfer routines are used, all the normal file transfer options apply. This allows a file to be transferred, retaining its file name on the local machine. The SmartEdit feature requires that the file ‘se’ be transferred to the host. Once it is on the host, just enter chmod +x se to enable it for operation.

SmartEdit uses the normal Windows extension support to determine which program to use to edit the file As with the first editing method, SecureNetTerm will examine the file length, date, and time to determine if the file was changed when the editing program exits.

WWW Browser Support

SecureNetTerm is designed to be the preferred telnet client of choice for www browsers. Select any telnet URL within a browser and SecureNetTerm will connect to that site. In addition, if you come across any type of URL while in a Telnet session, simply highlight the URL, then use the right mouse click, launch this URL option and SecureNetTerm will call your browser to connect to that URL

In order to define SecureNetTerm as the Telnet client for www browsers, use the SecureNetTerm Global Options-Applications-WWW Browser Telnet Handler option.

Supported Emulations

SecureNetTerm supports the following terminal emulations:

ANSI
ANSI-BBS
ANSI-CIS
BA-80
FTTERM
IBM-3101
IBM-3151
IBM-3161
IBM-3163
QNX2
SCO-ANSI
TVI925
VT52
VT100
VT102
VT220
VT320
WYSE50
WYSE60
XTERM

Keyboard Support

Overview

Keyboard definition for problematic keys is supported through a standard dialog panel. Selected keys, displayed within the dialog panel, can be programmed to transmit strings of your choosing. Each key can be set up to transmit one string when pressed by itself, a second string when pressed together with the <Shift> key, a third string when a key is pressed together with the <Ctrl> key, a fourth string when the Num Lock key is on, a fifth string when a key is pressed together with both <Ctrl> and <Shift>, and a sixth string when a key is pressed together with the <alt> key. The key definitions are stored concatenated together into a single string, whose total length cannot exceed 60 characters. The sub-strings are separated by the pipe character ('|', ASCII 0174). **Keys should not be defined when you are connected to a host**.

Special sequences such as escape, carriage return, and line feed are specified with the '^' symbol along with the printable 'equivalent' of the character. For example <Ctrl-c> would be encoded as ^C. To send the '^' symbol, you must escape it with the '\' character. The following are some additional samples:

^[	Escape
^M	Carriage Return
^J	Line Feed

Special sequences can appear at any point within the user-defined string for the specified key. For UNIX systems, the definition must match that within the corresponding TERMCAP or TERMINFO file. For dial-up lines, the keys can be defined to send any string. For example, you could define F1 to send the string 'ftp ftp.cica.indiana.edu^M', and F2 could be defined to send your Email address.

The Windows Common User Access: Advanced Interface Design Guide gives special meaning to the F1 key and the ALT key sequence. This presents programs such as SecureNetTerm with some design problems, since it must pass keys to another host, which in turn will process the keys according to their rules.

The F1 key is defined by the CUA guide as a user help key. SecureNetTerm will treat F1 as a user help key if it is not connected to any host; otherwise it will be passed on the host system as any other key. The ALT key sequence will normally be treated according to the CUA guide, that is these key codes will not be sent to the host. There are two exceptions to this. The first exception is for the twelve keys F1-F12. If these keys are defined with the ALT key modifier, then the defined data with is sent to the host. If the key is not defined, normal window actions will be honored. For example, ALT-F4 is normally used to terminate a program. If you define ALT-F4 using the keyboard dialog panel, then that definition will be sent to the host, and the normal Windows action will not take place.

If the first character of a key definition contains the "~" or "@" character, then the key is considered a local workstation key. A local workstation key can be used to select menues, or to run programs and scripts. The menu feature is enabled by prefixing a menu items label with the '~' character. For example, to select the Options-Certificates… menu item, use:

~Certificates…

The run feature is enabled by prefixing the full path of the program with the '@' character. This feature supports running programs or scripts. For example, to start the Windows Notepad program, the entry would be:

@Notepad.exe

Keyboard Definition

To setup custom values for keys, press the keyboard icon to bring up the keyboard dialog panel. Check to see if the keyboard definition you desire to change is loaded (lower left corner of the panel). If it is not, select the desired definition from the list of available definitions. To define a key, simply click the key with the mouse button and enter the new definition within the edit box at the bottom of the panel. After you have entered the new definition, press the "Change" push-button to make the change. If you want to define a Shift or Ctrl key sequence, click either Shift or Ctrl first, then click on the desired key. The Shift or Ctrl will stay on (as indicated by the buttons on the upper right of the panel) until you click on this again. The left Ctrl, Alt, Shift, and Caps Lock keys cannot be programmed. Normally, the only keys you should program are the twelve function keys. The most common reason to program any key is to reduce a long sequence of characters to a simple keystroke.

If you want to change the name of the keyboard definition, simply type over the current name. Once you have made all your changes, save the values by depressing the "OK" button. The new key definition will remain active until (1) you exit the program, or (2) you select a new host profile from the Profile Manager. If you wish to use the new key definition for a host profile, you must select that key definition in the Profile Manager. The purpose for keeping the new key definition active is to allow key changes while currently connected to a host, and then testing those changes while connected.

Accelerators

Since SecureNetTerm is designed to provide keyboard input for another host, the use of accelerators has been minimized to avoid conflicts with key definitions required by the host. The two exceptions to this are for the clipboard copy and paste functions. The accelerators Ctrl+Ins and Shift+Ins have been chosen for these. In addition, an option has been added to the Global Settings-Control dialog to turn on/off the use of accelerators. The default is off, which means the accelerators are not active. To allow the use of accelerators, simply select the 'Allow Edit Accelerators' option. A check mark will indicate that the option is turned on.

Numeric Keypad

The numeric keypad presents some additional requirements upon SecureNetTerm. The definition of the keys depends upon the state of the Num Lock key, and the type of emulation.

Note that the host under most conditions controls the numeric keypad.

Global Settings

Terminal

Options

The option "Bell on," allows for ignoring bell characters received from the host. The "Ignore Numeric Keypad" option allows SecureNetTerm to ignore host's requests to change the state of the keypad. This option is useful for portables, which do not have a normal numeric keypad.

The "Allow Program Calls" controls whether special host escape sequences can start a program on the users workstation. SecureNetTerm allows host applications to start programs on the workstation if this option is selected.

The "Allow Edit Accelerators" allow shortcut keys to be used for copy/paste options. Since SecureNetTerm is a terminal emulator, normally all the keys are defined by the emulation selected. However an exception can be made for the copy/paste operation.

The "DEC LK-450 Keyboard" option can be selected whenever a real DEC LK-450 keyboard is used on the workstation. This keyboard is designed after the real DEC VT series of terminals and contain special keys used by many DEC based hosts.

The "Right mouse click is copy/paste" option redefines the right mouse click default action. Normally, the right mouse click will bring up the right mouse menu. If this option is selected, the right mouse click will perform a copy/paste if text is highlighted on the terminal screen. If text is not highlighted on the terminal screen, the logic will then check to see if text is on the clipboard, and if so, will do a normal paste. When the copy/paste option is selected, a shift-right mouse click will bring up the right mouse menu. See the Mouse Selection topic for additional details.

The "Left mouse click sends row/column" option redefines the left mouse click default action. Normally the left mouse click will start a highlight text operation. If this option is selected, a left mouse click will send the current row/column location at the location of the left mouse click. This option should only be used with host applications designed to support this option, and it should be noted that this is not a standard feature of any known emulation. The selection of this option will also disable the double left mouse click hightlight word feature.

The "Ignore window title change requests" will prevent the host from changing the main window title, as well as the description on the window switching tab. Several host emulations, such as XTERM, have the ability to change the window title, often without the users specific knowledge.

The WAV File allows the selection of a Windows WAV file to be played whenever a file transfer completes.

The global download directory define where files transferred downloaded from the host will be placed. The global upload directory defines the default directory for files to be uploaded. Downloaded files will always be placed in the download directory. Uploaded files can be selected from any directory, but will default to the global upload directory.

Printing

SecureNetTerm has the ability to recognize the standard emulation escape sequences for printing. For VT-XXX emulation, whenever the escape sequence CSI[5i is received, a temporary file will be created and opened. Any data received after this point will be written to the file. When the escape sequence CSI[4i is received, the file will be closed. The following is an example of a UNIX script which will send the contents of a UNIX file to a local file for printing:

```
#!/bin/sh
echo '\033[[5i'
cat $1
echo '\033[[4i'
```

Data received after the transparent print option has been turned on will be captured in a temporary disk file. Once the printer data has been received and transparent printing is turned off, the file will be printed on your default Windows printer or processed according to the print option selected. Print options, such as the default font and font size for the printed file can be selected from Options-Global Settings-Printing menu item or from the toolbar. The temporary file which was created will be deleted by SecureNetTerm, if the 'delete file option is selected'. The following options control the handling of the printer data:

1. Receive File Only - Place the print data into a file.
2. Send to Windows printer - Send to the default Windows printer driver for processing.
3. Start user defined print program - Start the user defined print program, passing the file name.
4. Start user editor program and edit the file - Start the user defined edit program, passing the file name.
5. Add the print file directly to the Windows spooler queue - Add file directly to spooler queue.
6. Write direct to printer port .
7. SmartPrint - Queue all printout - Print on user demand.

Option five is designed to handle print files, which contain special printer control characters. This option should be used when printer formatting is done on the host. An example of this is a host-based application that creates a form, which includes laser jet print commands to set the font size, orientation, etc. Note that the File option 'Print Screen' is also controlled by the six print options, although option two is normally used.

Option 6 can only be used for printers connected direct using parallel or serial ports. This option cannot be used for printers connected using USB ports. Note that this option is not recommended unless option 2 or 5 cannot be used.

The SmartPrint option allows for the receipt of multiple print documents throughout the session to be placed within a common print file. The file can be printed, saved to another file, edited or deleted at any time.

The 'Display Print Setup at print time' option will display the standard Microsoft Print Setup dialog prior to printing, which allows for printer selection, paper orientation and other desired printer setup.

The file 'netprint', located within the SecureNetTerm directory, can be uploaded to your host using the rz command. Then issue the command chmod +x netprint. To print a file, simply enter netprint xxxxxx on the host, where xxxxxx is the file name to be printed. In essence, the netprint script can download any ASCII file to your machine. It passes all data received from the UNIX host to the local file, except for the binary zero character.

For all emulations other than VT-XXX, refer to the respective programmers manual for specific transparent printing escape command definitions.

Advanced

The advanced dialog allows for additional control over how transparent printing is handled, additional control of how the host bell character is handled and a couple of general items that some installations need.

When the current screen, or scroll buffer is printed, a decision must be made on how to handle special graphical characters. SecureNetTerm itself has the ability to display multiple fonts concurrently, which allows normal text and graphical characters to be displayed on the same screen. However, most printers do not have this ability. In order to display the text characters, and the line drawing characters, a printer font (such as the Microsoft Line Draw font) must be used. If this font, or similar font is not available or supported by the printer then SecureNetTerm can translate the line drawing characters to normal text characters.

Ansi Colors

Graphic attributes are the set of colors, which can be specified by the ANSI graphics attribute for background/foreground colors. This set contains sixteen colors, composed of the basic eight colors; black, blue, green, cyan, red, magenta, brown and white. Each base color is associated with the ANSI graphics color attribute zero through seven. Each of the base colors also has a corresponding high-intensity value that is used when the bold attribute is active.

Mouse Selection

The left mouse button is used to select text when the button is pressed, then dragged over text in the terminal screen area. SecureNeterm has two modes of text selection, as decribed in the text selection topic.

The middle mouse button is used to start mouse panning as described in the mouse panning topic.

The right mouse button can be used to bring up the right mouse button menu or to do a copy/paste. The default is to popup the right mouse menu. If the global option "right mouse click is copy/paste" is selected, the right mouse click will perform a copy/paste operation. In this mode, a shift-right mouse click will bring up the right mouse menu.

Locator Controller

SecureNetTerm supports the Locator Input Model for ANSI terminals (sixth revision). This document specifies the DEC application mouse support and the support for devices attached to the workstation serial communications port. The excellent program VTTEST, modified by T. E. Dickey, provides a test of the locator controller mouse support implemented by SecureNetTerm.

General

Applications

The applications dialog allows for the specification of the user preferred edit program, printer program and location of the Kerberos Ticket Manager. It also provides the ability to set SecureNetTerm as the Windows telnet handler for WWW browser support.

The printer program provides advanced printing support for those installations that have a need for advanced printing support, which is not supported by SecureNetTerm. If defined, SecureNetTerm will call this program when that option is selected in the printer setup dialog panel.

File Transfer

SecureNetTerm currently supports the Zmodem file transfer protocol.

The Zmodem protocol is by far the fastest file transfer protocol available. Numerous options are available, selected from the menu Options-Setup-Set File Transfer Options dialog panel. When sending files to a remote host, SecureNetTerm honors the following file management options:

ZMNEWL	Transfer file if destination file is absent, else overwrite if source is newer or longer.
ZMAPND	Append local file to the contents of the destination file.
ZMCLOB	Replace the destination file, even if it exists.
ZMDIFF	Transfer file if destination file is absent, otherwise replace if different length or date.
ZMPROT	Transfer file only if it does not exist on the host.
ZMNEW	Transfer file if destination file is absent. If present, overwrite if source is newer.

SecureNetTerm also supports 'crash recovery' with Zmodem. This simply means that if an error occurred while transferring a file, any subsequent transfers of the same file will proceed where the error occurred. This is extremely useful for long file transfers interrupted by loss of carrier or other errors. The option 'Do not buffer on send' should only be checked if you are having problems sending data to a host. SecureNetTerm normally buffers all outgoing data to achieve maximum transfer rates. On some hosts, buffering forces an overrun condition; this option should be used for these types of hosts.

ASCII Zmodem downloads are also supported (using the sz -a option on the host). If SecureNetTerm receives a download request which specifies the ZCNL option (convert received end of line to DOS end of line), all new line characters are preceded with a carriage return prior to writing to the local file. The ZCNL option is sent by the host when you request sz with the **-a** option. Please use this option with care: if the host file already contains normal DOS end of line characters, you will end up with a file that contains an extra carriage return. Also, since the host file was expanded in size (by the addition of the carriage return for each line), crash recovery will not work properly!

If the Zmodem option **sz -f** is used for downloads, the full path name must exist on the local machine. For example, if the command sz -f ~/work/file.c was issued at the UNIX host, it would be expanded by the UNIX host to its full path name. This could be something like /u/z/zkrr01/work/file.c on the UNIX system. On the local machine, the path /u/z/zkrr01/work must already exist or the transfer request will fail. Use this option with extreme care.

The 'block before write' option can be selected if overruns occur on file download. For SLIP/PPP and local Ethernet connections, this options has no meaning, since flow control is under the control of the TCP/IP protocol. For other types of modem transfers, it will correct most overflow conditions.

Under most conditions, the Zmodem options should be **no window, overwrite existing file always (zmclob)**.

Logo

SecureNetTerm supports the display of graphical images whenever a connection is not active. The Logo dialog allows you to enable/disable the logo display, select the transparent background color and the location of the graphical file to be displayed. Refer to the Logo How To for complete details.

Language

The language dialog allows the selection of the language to use within SecureNetTerm. Currently only English is supported.

Connection

Control

The retries, retry delay and connection timeout values control host connection attempts. In most cases, a connection happens immediately, however these values can assist with the connection of problem hosts/connections.

The "Maximum hosts in the quick connect history" controls how many quick connects hosts are maintained (remembered) by SecureNetTerm. The quick connect history is stored in the registry under the SecureNetTerm QuickConnect key.

The default user and password fields allow you to define a login userid and password on a global basis. SecureNetTerm will use these values if the userid and or password is left blank in the host profile. If the userid and or password is left blank in the profile and the default values are left blank, then they will be requested, if required, at connect time. The default user and password concept is useful for those installations that desire a common host file that should be identical for all users except for the userid and password. The user can then choose to use set the default values, set each host profile, or allow the program to ask for either.

Firewall

SecureNetTerm can support connections that require a firewall. Due to the nature and operation of firewalls, currently only FTP and SSH connections support a firewall.

Firewall options are global in nature. That is they apply to all profiles in the same manner. However each host profile has the option to turn off the firewall logic for that specific host. For complete operational details about your firewall or proxy server, please contact your system administration staff responsible for the firewall/proxy server.

The following are the valid firewall types:

Socks4	Normal Socks 4, can specify firewall userid.
Socks5	Normal Socks 5, can specify firewall userid and password.

X509 Server Validation

Server Validation

Server validation is a process by which the host presents its credentials to the client in the form of a certificate, and the client checks this certificate for validity to ensure the host that is responding to the connect request is in fact the host you want to connect to. The certificate presented, referred to as the server certificate, must have been created and signed by an CA (Certificate Authority). A copy of that CA certificate must be installed on your computer. The server certificate is first checked to make sure you do have a copy of the CA certificate, the server certificate has been signed by that CA certificate and that it is still time valid. The server certificate is then checked to make sure it has not been revoked, by checking the respective CA CRL (Certificate Revocation List) you have in your computers certificate store or that can be obtained in real time from distribution points contained within the server certificate or from external sources such as LDAP servers, and OCSP responders. If the server certificate passes all the criteria, the connection process will proceed. If not, you will be presented with a list of why you should not trust the host, and allowed to make the decision to proceed or disconnect. Once you make the decision to accept a hosts certificate, or SSH public key, it is normally saved in the known_hosts file for future connections

Server X509v3 certificate authentication is supported for the SSH-2 and TLS protocols. There are two independent models available, one Microsoft based, and the other based upon OpenSSL.

The Microsoft model allows for site specific requirements such as the ability to add revocation and LDAP providers to the CryptoAPI.

The internal OpenSSL based model has support internally for LDAP and OCSP CRL validation. These two models are mutually exclusive, that is, if the Microsoft model is selected, OCSP and LDAP support must be provided by user or company supplied plugins to the the Microsoft CryptoAPI.

If the internal model is selected, the CryptoAPI and any associated revocation plugins will not be utilized.

At the current time, the X509 Server Validation **does not apply** to the Globus authentication model. Refer to the Globus GSSAPI for details.

Options/OCSP

The Options/OCSP dialog consists of three areas, certificate validity options, certificate acceptance, and OCSP setup. The server validation model should be selected first, and can be either the Microsoft model or the internal OpenSSL model. If the Microsoft model is desired, select the "Use Microsoft CRL processing" option. If the internal OpenSSL model is desired, select the "Use internal CRL processing" option.

The "Enable endpoint identity check" option specifies whether the server's host name will be checked againt the servers certificate Subject Name or Subject Alternative Name fields. If this option is not selected, the host name will not be checked and validity will be based upon the certificate validity period and revocation check only. A possible security risk could result by not selecting this option, as anyone with a certificate issued by the same trusted CA that issued the host server certificate could perform a man-in-the-middle attack.

The "Disable CRL processing" option disables the certificate revocation list checks. Selecting this option is a security risk and **should only be used for testing**. Internally, all certificate revocation checks are done, but any resulting revocation is ignored. Results of intenal revocation checks are written to the internal log file. The internal log file can be configured/viewed with the programs Options-Show Log File menu option.

The "Use certificate distribution points" option instructs the internal crl logic to look for CRL distribution points in the server certificate, and if present, try to obtain that revocation information from the URL(s) specifed. If the Microsoft method is selected, the option is passed to the Microsoft CryptoAPI .

The certificate acceptance area allows you to provide a final check of the server certificate, after all the normal checks such as date, common name (CN) and revocation (CRL) checks have been made, by checking for a pattern in the server certificate subject line. If this option is selected, then normal known_hosts processing will be bypassed. The pattern can be against any of the fields contained within the certificate subject line, such as Organizational Unit (OU), Orgainzation (O), city (L), and state (S). During pattern matching, the certificate subject is processed with the following positional format:

CN=eagle.netterm.com, OU=Systems Development, O=InterSoft International Inc., L=Katy, ST=Texas

The pattern can check any field, or combination of fields, and can contain the normal glob characters "?" and "*". For example, if you want to check both the O and OU fields, then the following patterns could be used:

*OU=Systems Development**O=InterSoft International Inc.*

OU=Systems Development, O=InterSoft International Inc.

The Microsoft certificate manager (Options-Certificates) can be used to display the contents of a certificate subject, if you are in doubt about contents and positional order of the fields. The Microsoft certificate manager can also be activated with the current server certificate with the Options-Show Server Certificate if you are connected to the host. Note that each field is seperated by a comma, followed by a space.

The OCSP section provides for the ability to enable/disable the use of the internal OCSP client, and controls its operations. If the OCSP client is not desired, select the "Do not use OCSP for certificate validation" option. If the OCSP client is desired, select whether it should be utilized to validate all certificates, or just those that contain a valid authority info access url within the host certificate itself.

LDAP Servers

The LDAP Servers dialog provides for the ability to enable/disable the use of LDAP servers for the retrieval of certificate revocation information for host server certificate validation. If enabled, the location of these server(s) can be defined by a URL and the LDAP version to be used. Selection, modification and deletion of the LDAP servers is controlled by the Add, Edit and Delete pushbuttons. The order of the servers can be moved using the move up and move down pushbuttons. The LDAP control information is located in the SecureCommon.ini file, and is global in nature.

OCSP Responders

The OCSP Responders dialog allows for the definition of global responders to query for revocation information. The use of the responders is controlled by the options specified in the Options/OCSP section. The responders will be utilized in sequential order, so place the most used at the top of the list.

Responders are added by entering the responder URL in the URL entry/editing area, and pressing the add button. A responder currently in the list can be changed by clicking on the URL and pressing the edit button. Once it has been changed, press the add button to place it back in the list. Responders can be deleted by clicking on the URL, and pressing the delete button.. The order of the servers can be moved using the move up and move down pushbuttons. The OCSP control information is located in the SecureCommon.ini file, and is global in nature

Globus GSSAPI

Globus Configuration

Globus GSSAPI support is provided for SSH authentication, and is based upon SSL style certificate authentication. This style of authentication is normally supported only by SSH host servers that have been modified to support Globus related applications.

GSSAPI authentication is based upon a special short duration "proxy certificate" that is created from a user certificate. The proxy certificate is passed to the host during the connection process, and is used by the host server to grant access. The proxy certificate can also be used by the host to determine the login userid, if it is not specified by the SSH client.

In addition to the host verification process, GSSAPI specifications require the SSH client to verify the host certificate presented to the client during the connection process.

GSSAPI authentication requires a user certificate (usercert.pem) and associated certificate private key (userkey.pem) which is utilized by the SSH client to create the proxy certificate. In addition, the client host certificate verification requires a copy of the public portion of the CA certificate, which signed the host certificate, and the accompanying signing policy file.

These four files are normally created for a UNIX based user and placed within the users home directory in the .globus directory. The CA public certificate and accompanying signing policy file follow the naming convention <hash>.0 and <hash>.signing_policy, where <hash> is a value created by hashing the contents of the CA certificate. These two files are normally located in the .globus/certificates directory.

The four files should be copied from the UNIX host, and placed within a corresponding directory structure on the users Windows workstation. The selected location of the files can then be specified in the Global Settings-Globus GSSAPI Configuration panel. The location of the temporary proxy certificate can be left blank, which will allow the proxy generating process to place it in a user specific temporary directory. Once a temporary proxy certificate has been created, the location of the resulting certificate will be shown in the configuration panel.

The Globus GSSAPI also supports the placement of the user certificate, with its corresponding private key, as well as the public portion of the CA certificate to be placed within the Microsoft browser certificate store. Many organizations in fact issue the user certificate directly to the Microsoft browser, along with their CA certificate. If the certificates are currenly located in the Microsoft browser certificate store, then you can select the "User certificate is located in:

Browser" option located in the global Globus GSSAPI Configuration dialog. The only remaining file needed will be the CA Certificate signing_policy, which must be located in the directory specified by the "Trusted Certificate Directory". The required signing_policy file is often listed on the organizations web page responsible for the host you are connecting to, and can be downloaded from that location to the Windows workstation.

The generation of these four files is beyond the scope of this document. Please direct your inquires for the creation and location of these files to the authority responsible for the host you are trying to connect to.

CA Signing Policy File

Certificates bind a string (called a Distinguished Name) to a public key, together with some other data. The Distinguished Name (DN) is arranged heirarchically, much like a filesystem's directories and files are laid out.

Globus requires that CA certificates are accompanied by a signing policy, which specifies what subset of the CA Distinguished Name must be in the host certificate. The contents of a typical signing policy follows:

```
# EACL 1
access_id_CA   X509   '/O=Grid/OU=ThaiGrid/CN=ThaiGrid CA'
pos_rights     globus CA:sign
cond_subjects  globus '"/O=Grid/OU=ThaiGrid/*"'
```

The 'access_id_CA X509 line' is the printable contents of the CA certificate Distinguished Name. The Distinguished Name may contain an email address, which would appear as Email=myEmail@abc.com.

Normally, this would be at the end of the line, such as:

'/O=Grid/OU=ThaiGrid/CN=ThaiGrid CA/Email=myEmail@abc.com'

The attribute "Email" is dependent upon the version of Globus (or more precise, the version of OpenSSL utilized by Globus) that was used when the signing policy was created. On later versions of OpenSSL, the attrribute "Email" would appear as "emailAddress".

The Windows version of Globus GSSAPI uses the latest version of OpenSSL, thus whenever the Distinguished Name is converted to printable format, the "emailAttribute" attribute will result. If the signing policy file contains the attribute "Email", the host certificate will be rejected.

This is easily corrected by changing all references of "Email" to "emailAddress" in the signing policy file located on the Windows workstation.

Proxy Management

The Proxy Management panel displays the current proxy certificate, if any, and allows for the creation of a new proxy certificate. The key size of the new certificate and the length of time the certificate is valid can be specified.

Import Certificate Files

The Import Certificate Files panel controls a specialized utility designed to import OpenSSL style certificate files to the Microsoft certificate store. Although mainly used for Globus GSSAPI support, this utility may be used to import any RSA based certificate files to the certificate store.

Site Profile Manager

Profile Manager

The Site Profile Manager is composed of a site tree, containing site profiles and optional folders in which site profiles can be placed on the left side, and a detailed site profile on the right side. Note that the term "site" refers to an individual host, which can be connected to.

Each folder may contain multiple sites, which contain all the necessary information about how to connect to a single host and any necessary security control information and options. Each site profile must have a unique name, within a specific folder. When a new host is added, you should select the desired folder in which it should be placed, and a current host within that folder that most resembles the characteristics of the new host. Then press the “New Site” button on the lower left portion of the Site Profile Manager window. This will create a new site profile, containing default values. The new site profile can now be customized as required. If the Shift key is pressed at the same time as the "New Site", the contents of the selected host will be duplicated in the new site profile. This is a quick way to generate a new site profile for hosts with similar information. You can also drag a site from one folder to another can move folders and sites.

A site profile (the right side of the Site Profile Manager) allows you to enter the information required to connect and communicate with the host. The “Profile Description”, located at the top, allows you to enter descriptive information about the host, to help you remember the purpose and function of the host. The “Host” name can either be an IP address or a fully qualified network host name. The “Port” defines the host port to connect to and is dependent upon the connection type. The "Connection Type" list box defines what kind of software server you are using on the host. The “Interface Type” list box defines how you connect to the host, and in most cases it will be "Network". The "Modem" option is for those sites which is dialed direct by SecureNetTerm. This does not refer to a modem connection established by the Microsoft OS to your Internet Provider.

The "Exclude from firewall" exempts this host from the use of a global firewall. This is useful for those installations where the majority of their sites require the use of a firewall, therefore one is defined in the global options. For those hosts that do not require the use of the global firewall, selecting this option will bypass the firewall processing.

The "Default Site" option allows you to select the site profile which should be selected by the site manager each time the Site Profile Manager is started.

The "User" and "Password" text areas are optional, and are used to supply the host with your userid and password. If these are not supplied, and they are required by the host for the login process, SecureNetTerm will request them during the connection process.

If you place a password in the profile, it will be saved in the SecureCommon.xml file, however, this violates most company security policies and should not be used if your company has such a policy. If a password is not saved, SecureNetTerm will request it when needed. Passwords saved to the SecureCommon.xml file are encrypted with a very simple encryption method, and should NOT be considered secure.

For telnet based hosts, which required a script for logging in, the "Login Script" text area allows you to specify the script file to be used. Refer to the section ActiveX scripting for script details.

The "Mapping File" text area allow for the definition of a language translation file. See the section on language considerations for more detailed information.

Tools

Import

The import tools allow you to import selected items from your NetTerm or SecureNetTerm version 5.4.3 and previous control files. NetTerm maintains its global data and phonebook in the netterm.ini file. SecureNetTerm maintains its global data and phonebook in the netterm2.ini file. The Import Globals tool imports items of global nature from these files. The Import Phonebook tool will import sites (hosts) from these files and place them in the current Site Profile Manager .xml file. Note that you must select the 'Save and Exit' or 'Connect' option when you exit the Site Profile Manager to make the import permanent.

Site File Management

The Site File Management tools allow you to import/export selected folders from the current Site Profile Manager .xml file. The high level folder to be exported should first be selected (highlighted) in the folder/site tree. The export tool will save all folders/sites that is a "child" of the selected folder to the file that you specify. Note that the purpose of these tools is not for backup. They are designed to allow selected folders of your site management files be exchanged with others.

In addition to these tools, you can specify the Site Profile Manager .xml file to use on the command line or change the global Site Profile Manager .xml file in the registry.

Dual Use

SecureNetTerm, as well as SecureFTP, has the ability to share a common Site Profile Manager .xml file. Profiles within that file can be (1) SecureFTP only, (2) SecureNetTerm only, and (3) Dual use. The Dual use function allows a single site profile to contain common information about a single host which can be used by both programs. Dual use profiles should only be used when needed, since both the size of the .xml file is affected, as well as the load time for the Site Profile Manager. Profiles which are only used for FTP access, should be declared as SecureFTP style profiles. The same holds true for hosts which are only accessed by SecureNetTerm.

Advanced Host Settings

Terminal Settings

DeskTop

The Desktop dialog allows you to control host-dependent display information such as the number of rows displayed and the number of columns. Most terminals have 24 rows by 80 columns in normal mode and 24 rows by 132 columns in report mode. However, some host applications have a need for more, so you are given a choice. All this really controls is the screen model for those applications, which create menus and expect to have a defined number of lines. Most applications simply write lines, thus the screen will scroll whenever the maximum number of lines have been displayed. The same is true for the maximum number of columns, which can be displayed.

SecureNetTerm supports the **NAWS** option, which will allows the terminal to send the current number of rows/columns to the host. This option is defined within RFC-1073 and is supported by the most Telnet and SSH servers. When used in conjunction with the desktop number of rows/columns option, larger screen sizes can be obtained. Refer to the 'resize' command on your UNIX host for additional information on the use of larger screen sizes.

The return sends, line control, and scroll-back options further allow you to enhance the display. The 'return sends' option determines the type of end-of-line sequence to send to the host. Local echo determines how characters typed locally are treated. If characters typed on the local keyboard do not appear on the screen, it probably means that you need to select this option. The auto-wrap options controls what happens whenever a line sent from the host exceeds the number of columns you have selected. If auto-wrap is on, any characters received after the maximum have been exceeded will be displayed on the next line. If it is off, each character received after the maximum has been reached will be displayed in the last column.

The 'Time Format' options controls what is displayed in the time field on the status bar. The options are elapsed time, current time or the current row/column location of the cursor.

The option to turn off the blinking cursor should be used with caution. The cursor is common to all Windows applications, so if you turn off blinking within SecureNetTerm, it will turn off blinking in all applications. The option will be valid only when SecureNetTerm is connected. The cursor blinking rate will be set to its original value when SecureNetTerm is not connected. The preferred method to set the cursor blinking rate is within the Microsoft control panel.

The “Enable UTF-8 host character encoding” option turns on host UTF-8 support. This option can be used if your UNIX host supports UTF-8 and it has been enabled for your login profile. If the UNIX command:

```
locale charmap
```

results in a reply of UTF-8, then it is enabled.

UTF-8, when used with a sufficiently populated monospaced font (such as Andale Mono, Everson Mono Unicode or Courier New) can display text in many languages and writing systems within the same screen. The "UTF-8 and Unicode FAQ for Unix-Linux" by Markus Kuhn is an excellent source of information on UTF-8.

The scroll-back option allows you to define the number of lines that are kept in the scroll-back buffer. As lines get scrolled off the screen, they are transferred to this buffer, up to a maximum of 32,767 lines. The vertical scroll bar will use the number of lines contained within the scroll-back buffer to control the relative position of the scroll button.

The scroll slow option allows you to set the rate at which lines are placed on the screen. If this option is not selected, lines sent by the host will be placed on the screen at the fastest possible rate. If the option is selected, you should enter the rate (in milliseconds) at which the lines should appear on the screen. The default rate is 100 milliseconds.

The 'Answer Back' option provides a method to identify your terminal to the host system. The contents of this text field will be sent to the host when it is requested.

The 'Terminal Type' field allows you to override the terminal identification string sent to the host when requested. If this field is defined, it will override the default terminal type information associated with the emulation selected. For example, if VT220 emulation is selected in the Site Manager for a host, SecureNetTerm will send the string "vt220" as the terminal type. If you set the 'Terminal Type' field to "vt200", that value will be sent instead of "vt220".

Window Sizing

The Window Sizing dialog provides for the selection of how SecureNetTerm should handle the sizing of the window by the user, or by the host.

When the window is sized by the user, by grabbing a portion of the window with the mouse, then dragging it to increase or decrease it size, you can (1) change the number of rows/columns and keep the font size constant, or (2) Change the font size and keep the number of rows/columns the same. Selection two is the preferred method.

When the window is sized by the host, such as switching from 80 columns to 132 column report mode, SecureNetTerm offers three methods to deal with the window size. The first is to keep the same font and font size, and employ horizontal sizing. The window size remains the same. The second is to keep the same font and font size, but the window size is changed to match the additional columns. The third, and preferred method, is to keep the same window size, do not use horizontal scrolling, and scale the font to current window size. This method has one additional option to maximize the number of rows displayed.

QuickButtons

QuickButtons provide a quick and easy way to send keystrokes to the host using the mouse. The buttons are located directly below the toolbar and can contain up four sets of eight buttons. The normal QuickButton bar displays one set of eight buttons. The active set can be selected by a mouse clink on the "Change QuickButton Set" icon located to the left of the buttons. Each mouse click will select the next sequential set. If desired, all four sets can be displayed at the same time by selecting the View-Use QuickButton Pad. The QuickButtons can be enabled/disabled on a global basis within the View menu.

Each button can be defined with a button label and up to 255 characters which can be sent to the host system when pressed. Control characters such as carriage return can be a part of the string and follow the same conventions as defining keys within the keyboard dialog box. An example of the use of QuickButtons to perform several commands is:

```
cd test^Mls -l^M
```

QuickButtons also supports URL's, menu items, and starting other Windows programs, providing a quick and easy way to access commonly used menu items, run programs, and control a complex program startup on the host.

If an html style URL is detected, SecureNetTerm will start the users preferred browser and pass it the URL. This is a very powerful feature, allowing unique uses of SecureNetTerm with a browser. For example, you could use this feature to start an ftp request to the host:

```
ftp://user@myhost.com
```

The menu feature is enabled by prefixing a menu items label with the '~' character. For example, to select the Options-Certificates… menu item, use:

~Certificates…

The run feature is enabled by prefixing the full path of the program with the '@' character. This feature supports running programs (.exe or .lnk), or scripts. For example, to start the Windows Notepad program, the entry would be:

@Notepad.exe

Labels can contain any number of characters, but keep in mind that the size of the buttons will vary with the screen width. Once any button is defined, the bar will be displayed each time the host is selected. To remove the QuickButton bar, simply remove the definitions from the QuickButton dialog box. A unique QuickButton bar can be defined for each host.

Up to four unique sets of QuickButtons can be defined, for a total of 32 buttons. If the global menu View-QuickButtons is enabled, and QuickButtons have not been defined for the active host, the default QuickButtons will be used. Default QuickButtons are defined in the SecureCommon.ini file the section key DefaultQuickButtons. The use of the default QuickButtons is an excellent choice for those that desire the same set of QuickButtons for all or most hosts.

The "Clear all" option will clear all QuickButton definitions, for all sets. The "Save to file" option will save all three sets to a user specified file. The "Load from file" option will load all three sets from a user specified file. The "Save as default" option will set the defaults contained within the SecureCommon.ini file to the values contained in the current three sets.

The left edge of the QuickButton bar contains two control icons. The first will bring up the QuickButton dialog allowing for defining/changing the QuickButtons. The second control icon will switch the QuickButton display to one of the other two sets of buttons.

Screen Colors

Color within the SecureNetTerm environment is divided into two main areas when connected to a host, text and the ANSI graphic attributes. The main text attributes are normal and reverse video. SecureNetTerm will also allow changing the colors for blinking, bold; underline and line draw characters, although this is not normally done.

The text attribute colors can be controlled, by host, within the Advanced Settings-Terminal-Screen Colors dialog. To change a color attribute, select the radio button corresponding to the text color you desire to change, then change to the desired color in the color box to the right of the attributes. The sample screen at the bottom of the color dialog panel will provide a preview of what the new color will look like.

When SecureNetTerm is not connected to a host, the background color of the host is controlled by the toolbar background color icon. SecureNetTerm will also display a "Logo" within the main screen area. The display of the icon can be turned off in the Global Settings-Terminal-Advanced-Disable Screen Logo dialog item. The "Logo" displayed is controlled by the "Logo" key located in the Global registry folder for SecureNetTerm.

Extended Options

The extended dialog provides special control features required for some hosts.

The 'Add LF to received CR' and ' Add CR to received LF' can be selected if the host does not provide the normal CRLF sequence that SecureNetTerm expects at the end of each line.

The 'Exit on disconnect' option provides a quick method of quitting SecureNetTerm any time the line connection has been closed. When this option is selected (checked), SecureNetTerm will exit whenever the 'disconnect icon' is pressed or when the host has closed the current connection as a result of the log off command. Host timeout disconnects will also cause SecureNetTerm to exit.

The 'Do not disconnect if session active' options prevents closing the connection as long as you are logged into the host. The option is helpful for those sites that require an orderly shutdown of applications such as databases.

The 'Select linemode for local input' is required for some 'Talker' applications, and connection over a radio link.

The National Character Set option allows for selection of the desired National character set to be used by the connection.

Window Options

The window options dialog provides the ability to define the control bars, window title and color scheme by host. The control bars and color scheme are normally defined on a global basis (applies to all connections), and are set in the View menu. The window title is normally the host profile name.

If there is a need to customize the control bars, color scheme, or window title on a selective host basis, this dialog provides the ability to do so. To customize, simply select the control bars and or color scheme desire, then press the corresponding set button. The window title can be set by entering it into the optional window title field.

These options will take effect the next time the program is started for that host. You can remove the host specific options at any time by selecting the reset button for the control bars and color scheme, and by removing the optional title.

Modem/Direct Connect

In order to dial a system direct, such as a bulletin board, you must first define your modem to SecureNetTerm. If you do not connect to these types of systems, you can ignore this section.

To set up the modem and corresponding communication port, open the Site Manager and select the Advanced Settings-Modem/Direct Line dialog. This dialog panel is composed of two major areas, the "Communication Line Options" and the modem selection.

In order for SecureNetTerm to communicate with your modem, you must define the communication port, baud rate, data bits, and parity and stop bits for the serial port. Next set the "Control Types". Normally these are set to "RTS/CTS", and "Tone dialing". If you are connecting direct to a host (no modem) select the "Direct Line" option.

If you are using a modem, you must select the modem from the "Connect Using" area. Be sure your modem is displayed in the selection text box to the right of the Modem label. If not, left click on the drop down arrow and select the modem.

Caution: Very few sites today support direct modem connects. If your computer system uses a modem for an Internet connection, it cannot be used by SecureNetTerm if it is currently in use for that purpose. **The only people that can tell you how to connect to their host, and the required communication setup, are those responsible for that host.**

Session Logging

Session logging captures data received from the host to be written to a user selected disk file, providing an audit trail of the session. The log file name can be a fixed name, or it may contain variables which are determined when the file is first created. The configuration dialog lists the available variables, and a "Resolved log file name" text box which will display how the filename will be formed from the specified variables. The option "Enable Logging" controls if the log file is to be active during the current session.

Logging can be toggled on/off during the session by selecting the View-Session Logging menu item. Logging data can be appended to the current file, or you can select to overwrite the data currently contained within the log file.

Note that information received from the host in full screen mode may not be fully captured. The logging facility depends upon physical lines being sent from the host. Many menu based programs operate in full screen mode which do not have physical lines, that is, the data is written at random at whatever row/column position the cursor is moved to.

SSH

The SSH protocol provides strong authentication/encryption to communicate with the host SSH server. The SSH1 and SSH2 protocols are derived from the OpenSSH project and session encryption is provided by the OpenSSL project.

SecureNetTerm will attempt to use the SSH-2 protocol by default, however if it detects a SSH-1 style server, it will attempt to start the SSH server using that protocol.

The SSH protocol contains quite a few options, such as compression, encryption cipher to use, and authentication type. The Security Manager allows you to customize the SSH options to achieve optimum performance and to work properly with your host SSH server. This manager will also allow you to generate public/private keys and certificates.

Authentication

The SSH Authentication panel allows you to select the method of authentication to be used to prove your identity to the host. SecureNetTerm supports the following authentication methods:

SSH1, SSH2	Encrypted password
SSH1, SSH2	RSA public/private key
SSH1, SSH2	Challenge/Response such as S/Key and OPIE
SSH2	GSSAPI-Globus
SSH2	GSSAPI-Kerberos
SSH2	Hostbased (Rhosts with RSA)
SSH1	Rhost/Rhosts with RSA

If the GSSAPI-Kerberos authentication method is used, the workstation must have either the MIT Kerberos support installed, or be a member of an Microsoft domain which has SSPI support. The "Microsoft SSPI Server Realm" input area allows you to specify the UNIX Kerberos realm, if needed. The GSSAPI-Globus and GSSAPI-Kerberos authentication will use the external-keyx method if the userid is not specified. The userid is derived from the user certificate for globus, and the kerberos ticket for Kerberos. The external-keyx method has the added advantage of not requiring the use of the known_hosts processing to verify the host.

If hostbased authentication is selected, then the key to be used must be selected in the "Rhosts-host Key Authentication" section. The selected key can be a private key or a public key. If a private key is selected, it should be encrypted and will be processed directly by SecureNetTerm. If a public key is selected, then SecureNetTerm will attempt to use SecureKeyAgent to perform the authentication. This allows SecureKeyAgent to manage the private keys whether they are disk based, contained within the Microsoft browser, or located on a smart card or usb token. The public key feature is only available with the SSH2 protocol.

Key Management

Public key authentication uses a public-private key pair to login to host SSH servers. These key pairs can be a certificate located in a browser certificate store, or a disk file containing the private and public key in a unique format. A certificate is composed of two parts, the certificate itself containing a public key and related informational data, and a private key. Both will be referred to as simply a key pair or identity files.

SecureNetTerm supports private keys stored within a disk file, controlled by SecureKeyAgent, or a private key associated with a certificate in the Microsoft certificate store.

Internally, private keys are stored in the OpenSSH private key format. However, private keys generated by the Putty and SSH Data Communications (SSH.com) programs are supported. These key formats can also be converted to the OpenSSH private key format.

Creating/Installing Public/Private Keys

To generate a new key pair, select the SSH-Key Management panel, then make sure you have the SSH Key File option selected. Then press the Generate Keys button. This will start the Security Wizard, which will ask where you want to save the key pair, then will lead you through several steps to generate a new key pair. Once the key has been generated, you have the option to enter a passphrase, add a comment and provide the filename in which to save the public and private keys.

Exporting the Public Key

Once you have created your key files, there are several steps that will need to be completed to make use of them with SecureNetTerm. The necessary steps are 1) export the public key to a disk file, 2) transfer the public key file to the host and 3) configure the host SSH server to recognize your public key.

To export the public key, select the "Export Public Key" button. You will be presented with a dialog requesting the format for the public key file, the file name for the public key, and an optional comment. The format is server dependent, so be sure to select the one that corresponds to the host server type. The two most popular are the OpenSSH server and the SSH Data Communications server. If in doubt, contact the system administrator of the host you will be connecting to.

Then press the "Export" button to save the public key to the specified filename. Once the key has been saved, proceed to the advanced section on configuring the host server to use your public key.

The public key will also be placed within the text box labeled "Public key for pasting into host OpenSSH authorized_keys file". It then can be copied to the clipboard, providing the ability to pasting it directly into the host authorized_keys file (if you are online).

Once the keys have been generated, you can export the public key at any time.

Parameters

SSH Parameters allow you to specify the preferred cipher and preferred cipher order, along with the SSH-2 HMAC and compression level.

SSH allows for the mutual selection of ciphers to be used to encrypt/decrypt the connection to the host. In those cases where the host supports the same ciphers as SecureNetTerm, your preference will be honored. SecureNetTerm supports the most popular and widely used ciphers, including the AES cipher suite. The "Use default ciphers" is the recommended way to select ciphers, allowing future ciphers to be utilized automatically.

Data stream compression is defined within the SSH-2 specification, but some SSH servers have trouble with this option. If you have trouble connecting to a SSH host, turn on the logging feature and attempt to login again. Then review the log file and if you see a message stating referring to compression failure, or cannot find a suitable “comp” method, you will have to set the SecureNetTerm compression level to 0 (None).

Forwarding

Port forwarding is the ability to secure TCP/IP traffic using SecureNetTerm's SSH1 and SSH2 protocol support. This means that you can encrypt application data using protocols such as POP3, FTP, and SMTP. For example, if you receive your email from an Internet Service Provider (ISP), you could encrypt the communication between your email client on the local workstation and the ISP's SSH server. SecureNetTerm also supports X11 forwarding, which allows X Windows traffic between the X server and X client to be encrypted. In general, with any port forwarded by SecureNetTerm for an application, the application needs to be configured to use the localhost or loopback address 127.0.0.1 as its application server address.

The "Port Forwarding" section allows for the definition for both local and host ports to be forwarded.

The "X Forwarding" option, if selected, will setup the proper forwarding for your Windows based X Server. Once the X Server has been started, and an active SSH connection has been established with the host, you can display host X applications on your workstation. Note that SecureNetTerm is not an X Server. The "X Forwarding" option allows SecureNetTerm to accept X11 data from the remote machine and forward it to the X Server running on the local machine. The local X Server must be running before any X11 sessions can be displayed.

The "FTP Forwarding" options defines the proper port forwarding for support of an external FTP client. If your FTP client supports PASV for the data ports, select the "FTP Data Port 20 (PASV)" option in addition to the "FTP Command Port 21" option. The use of an external FTP client which supports PASV on the data ports, with port forwarding will provide secure ftp data transfers.

Known Hosts

The SSH protocol provides for the ability to identity a host by a unique key. The first time you connect to a host, that host offers a unique key. If the unique key is accepted, the key is saved in the known hosts file. For each connection after the first, SecureNetTerm will check the key presented by the host and checks to see if it is contained within the known hosts file. If the key is present, and matches the one contained within the file, the connection is accepted without further user intervention. If the key does not match, the connection will be terminated, or a warning dialog to presented indicating a possible security problem.

The strict host key checking option allows for the selection of how SecureNetTerm should handle host key changes. If strict host key checking is selected, SecureNetTerm will automatically disconnect if it receives a key from a host that does not match the host key contained within the known hosts file. If it is not selected, a warning dialog will be displayed.

The host unique key contains a large number of digits, making it very difficult for a user to check. For this reason, SecureNetTerm presents a "key fingerprint" of the unique key to the user for comparison and acceptance.

At times, the host administrator may generate a new unique host key. If so, anyone that had connected to that host and accepted its prior key, will be disconnected, or receive the warning dialog about a possible security problem. Normally the host administrator will notify all of the users of the system about the key change. In this situation, the old key can be deleted from the known hosts files by highlighting the key and then using the "Delete" button to remove it from the file.

When SecureNetTerm is initially installed on a user workstation, an empty known host file is created in a user dependent directory. The location of this file will not be displayed in the known hosts dialog, since it is the default file. If desired, you can select a new known hosts file/directory.

SecureNetTerm maintains host keys within the known host file by the host IP address and by port number. This allows those installations that have a need for multiple SSH servers on the same host to have a unique key by port as well as by IP address.

Key changes normally represent a very serious security problem and should not be treated lightly. They should be reported to your companies security department immediately.

Globus

SSL/TLS

Client certificates, if used, or required by the host SSL server, can be defined with the Advanced Host Settings Manager. If a client certificate has not been specified prior to the connection attempt, and the host requires the use of a client certificate, an interactive display will be presented to the user requesting the certificate to use. At the current time, only Microsoft RSA certificates are supported. These certificates can be within the internal Microsoft certificate store, or located on any Smart Card/USB token supported by Microsoft.

SSL/TLS Authentication

The Authentication dialog allows for the selection of an authentication method. The most common SSL/TLS authentication method is the certificate. The kerberos 5, srp, and password authentication methods are also supported for those with special or unique needs.

Client certificates must reside in the Microsoft certificate store. The Microsoft Internet browser is normally used to maintain, import and export client certificates as well as create certificate requests which can be signed (approved) by your companies CA authority, or one of the several commercial CA authorities. In addition, SecureNetTerm can generate client certificates (using the Options-Create User Certificate menu item) using your companies CA certificate.

You can also display and export the client certificate in this panel. The export function will create a disk file of the certificate, which is required by most UNIX hosts.

Creating/Obtaining a Certificate

To obtain an X.509 certificate, begin by generating a public/private key pair on your client machine (refer to your system documentation for information on this process). Keep your private key on the local machine and forward the public key with any other required information to a CA (Certificate Authority) in the form of a request for certification. If the request is approved, the CA digitally signs the certificate and returns it to you in the form of a certificate. Companies using browser related systems normally enhance this process.

SecureNetTerm includes the ability to create client authentication certificates for those that have a valid CA certificate (Options-Create User Certificate).

Once you have generated/obtained your certificate, start the Site Profile Manager, select the host the certificate will be associated with, then press the "Security" button. In the Security Manager, select the authentication panel, "Browse" button. You will then be presented with a list of certificates contained within the Microsoft user certificate store. Select the certificate you have just generated/obtained. Then proceed to the section titled "Exporting the Public Key".

Exporting the Certificate

Once you have created your certificate, there are several steps that will need to be completed to make use of it with SecureNetTerm and the host server. The necessary steps are 1) export the certificate to a disk file, 2) transfer the certificate file to the host and 3) configure the host server to recognize your certificate.

To export the certificate, start the Site Profile Manager by pressing the "Site Profile Manager" button on the toolbar, select the desired host, and then press the "Advanced" button. This will start the Advanced Host Settings Manager.

The current client certificate will be displayed in the “Authentication Certificate” display area. When you press the “Export Certificate” button, the Microsoft Export Certificate Wizard will assist you in exporting the certificate. Select the options (1))No, do not export the private key, and (2) Base-64 encoded X.509 (.CER) format. Once the certificate is exported (be sure to note where you told the Wizard to place this file), transfer the file to the host and use any editor to place within the .tlslogin file.

SSL/TLS Ciphers

The Ciphers dialog allows for the selection of a specific cipher to be used for data traffic encryption/decryption. Under normal conditions, the selection of a specific cipher is not needed, or recommended. The selection of a specific cipher may be needed for internal company policy or for interfacing with specific SSL based servers that support a limited number of ciphers. The choice of ciphers available is dependent upon the authentication method and whether both stong and week (export grade) ciphers are allowed. For maximum security, you should always select the strongest cipher available.

The selection of a cipher is a mutual agreement between the client program, and the host server program. That is, if you select a cipher that is not supported by the host server, the connection cannot be established.

ActiveX Scripting

Creating Scripts

SecureNetTerm is capable of hosting ActiveX Script engines. The most common ActiveX script engines are VBScript and JScript (Microsoft's version of JavaScript), both of which are freely available from Microsoft. Chances are you already have them installed if you've installed later versions of the Internet Explorer.

ActiveX script engines communicate with SecureNetTerm via standard interfaces. Therefore, SecureNetTerm can host any compliant script engine to run your scripts. The advantage of this approach is that you can script SecureNetTerm using the language of your choice. If an ActiveX script engine is available for your preferred scripting language, you can write scripts that will work with SecureNetTerm.

Scripts are text files that you create with your text editor, or the builtin script editor. Each script that SecureNetTerm runs must have a header that begins on the first line of the script. The header is used by SecureNetTerm to identify which script language the script is written in and the version of SecureNetTerm's scripting interface. Each line of the script header must begin with a (#) character. The SecureNetTerm script header requires a **$language** line that identifies the script engine and an **$interface** line to identify SecureNetTerm's interface version.

The syntax of the script header is always the same regardless of the script language you are using.

A simple but complete SecureNetTerm script with a header that identifies it as VBScript is shown below:

```
# $language = "VBScript"
# $interface = "1.0"
Sub Main
  ' Display Message
  snt.MessageBox "SecureNetTerm is at your command!"
End Sub
```

Note: Comments contained within scripts must be in the format defined by the script engine.

The quoted string following **$language** identifies the script engine. The following language keywords are supported:

Microsoft Visual Basic Scripting	VBScript	.vbs
Microsoft Java Script	JScript	.js

ActiveState Perlscript	Perlscript	.pl
www.python.org with Mark Hammond's win32all Extensions	Python	.py

Currently the script header should specify version 1.0 for **$interface**. Future versions of SecureNetTerm may support other versions. The example script above has a subroutine named main where all of the script's code is located. When SecureNetTerm executes scripts it always attempts to run a main routine if you have defined one.

It is not a requirement that you place your code within a main, however there may be reasons why you would want to do this. The VBScript and JScript engines will parse and execute global script code (script code you have defined outside of any subroutine) before your main is executed. If you have "initialization" code that you want to ensure has been completely executed before your actual script code begins, it may be useful to place your initialization code at the global level. This will ensure that your initialization code will all execute before your main code runs.

Another reason you may want a main routine is to allow your scripts a way of aborting themselves in case of problems. In VBScript there is no built-in way of exiting a script at the global level. However, if you want to exit a subroutine it is possible to use the Exit Sub syntax to do so. For example, in VBScript:

```
Sub Main
  condition = DoSomething()
  If condition = 0 Then
    ' Error, bailout
    Exit Sub
  End If
 End Sub
```

When the main routine ends the script has finished running. By placing your code within a main you have the option of invoking Exit Sub whenever it might be necessary.

The previous script samples are written in VBScript. The remainder of code samples in this document are all written in VBScript unless it is stated otherwise. The properties and methods of SecureNetTerm's interface can be used as documented by any compatible scripting language.

Script Editor

The Script Editor provides editing and testing support for the development of scripts. The Scintilla editor provides extensive editing capabilities, as documented online.

Autocompletion of the SecureNetTerm properties, methods and events is provided; triggered by the **snt. snt_** or **snt->** keywords. When one of the two keywords is detected, a popup window will be displayed containing all the methods/properties or events. Each additional character typed after the keyword will be used to select the closest match. Double clicking of an item in the autocompletion list will close the autocompletion window and place the selected item in the editors window.

Handling Script Errors

When scripting SecureNetTerm with VBScript, there are a couple of possible ways of dealing with script errors. For simple scripts, one possibility is to assume that the script will work properly, and not deal with runtime errors that may occur.

For more robust error handling a SecureNetTerm script should use the VBScript statement On Error Resume Next. Placing this statement at the top of a routine tells VBScript not to terminate the script, but to set the ‘Err’ object and continue execution. This makes the script responsible for handling failures. In the event of a runtime error, scripts that handle these errors should check the value of VBScript’s ‘Err’ object after calls to SecureNetTerm functions that may fail.

SecureNetTerm will attempt to display scripting errors detected during compilation, and will display the error messages generated by the script compiler. Note that the required lines:

```
# $language = "VBScript"
# $interface = "1.0"
```

are removed by SecureNetTerm prior to passing the script to the script compiler. Therefore, line numbers reported by the compiler may be off by two for those scripts that contain these lines.

Tips/Tricks

By calling WSH-VBScript functions, you can achieve some useful tasks very easily. The following example demostrates how to call the Windows RUN dialog:

```
Sub Main()
Set wsh = CreateObject("Shell.Application")
wsh.filerun
End Sub
```

The following example shows you how to start the Windows Calculator and use the function "SendKeys" to control it.

```
'*********************************************************************
' SendKeys can send "special key" using the following code :
' BACKSPACE {BACKSPACE}, {BS}, or {BKSP}
' BREAK {BREAK}
' CAPS LOCK {CAPSLOCK}
' DEL or DELETE {DELETE} or {DEL}
' DOWN ARROW {DOWN}
' END {END}
' ENTER {ENTER} or ~
' ESC {ESC}
' HELP {HELP}
' HOME {HOME}
' INS or INSERT {INSERT} or {INS}
' LEFT ARROW {LEFT}
' NUM LOCK {NUMLOCK}
' PAGE DOWN {PGDN}
' PAGE UP {PGUP}
' PRINT SCREEN {PRTSC}
' RIGHT ARROW {RIGHT}
' SCROLL LOCK {SCROLLLOCK}
' TAB {TAB}
' UP ARROW {UP}
' F1 {F1}
' F2 {F2}
' F3 {F3}
' F4 {F4}
' F5 {F5}
' F6 {F6}
' F7 {F7}
' F8 {F8}
' F9 {F9}
' F10 {F10}
' F11 {F11}
' F12 {F12}
' F13 {F13}
' F14 {F14}
' F15 {F15}
' F16 {F16}
' SHIFT +
' CTRL ^
' ALT %
'*************************************************************************

Sub Main()
Set wsh = CreateObject("WScript.Shell")
Set sink = CreateObject("EventMapper.SecureNetTerm")
wsh.Run( "calc")
sink.Sleep (100)
wsh.AppActivate( "Calculator")
sink.Sleep (100)
wsh.SendKeys( "1{+}")
sink.Sleep (500)
wsh.SendKeys ("2")
sink.Sleep (500)
wsh.SendKeys( "=")
sink.Sleep (500)
wsh.SendKeys( "*4" )
sink.Sleep (500)
wsh.SendKeys( "=" )
' 1+2 = 3  * 4 = 12
End Sub
```

SecureNetTerm ActiveX Object

Window Control

Description

The Window related properties and methods provide access to SecureNetTerm's window such as the window’s visible state, caption, etc.

Window Related Properties and Methods

Properties	Methods
Visible	GetViewWidth
Caption	GetViewHeight
WindowState	StatusLine
Batch	SetStatusLed
	QuitApp
	CreateActiveX

Visible

Description

Returns the state of the main SecureNetTerm window. True if visible, otherwise False.

Caption

Description

Returns or sets the title or caption of SecureNetTerm's application window.

Syntax

object.Caption [= *string*]

WindowState

Description

Returns or sets the state of SecureNetTerm's application window.

Syntax

object.State [= value]

Remarks

The state may be one of the following values:

0 - hidden
1 - visible (normal)
2 - minimized
3 - maximized

Batch

Description

Returns or sets the state of the SecureNetTerm batch mode. If in batch mode, interactive messages will not be displayed. Messages displayed from the script itself are not affected. True if in batch, otherwise False.

GetViewWidth

Description

Read only method to return the current width (in pixels) of the SecureNetTerm window.

Syntax

Width = object.GetViewWidth

GetViewHeight

Description

Read only method to return the current height (in pixels) of the SecureNetTerm window.

Syntax

Height = object.GetViewHeight

StatusLine

Description

Writes a message on the SecureNetTerm status line.

Syntax

object.StatusLine("message")

SetStatusLed

Description

Sets/resets the status line LED indicators.

Syntax

object.SetStatusLed(1)

Remarks

The indicators are arranged on the status line, numbered from left to right as 4, 3, 2, 1 and are represented by a single hexadecimal encoded decimal value, which allows for all LED(s) to be set/reset with a single digit. To set all the indicators, a value of 15 would be sent. To set indicator 1, a one would be sent. A value of 12 would be sent to set indicators 4 and 3.

QuitApp

Description

Closes the SecureNetTerm application.

Syntax

object.QuitApp()

CreateActiveX

Description

Creates a new object.

Syntax

Dim word

Set word = snt.CreateActiveX("Word.Application")

Session Control

Description

The Session related methods and properties provide access to the host state.

Remarks

SecureNetTerm's Session object is accessed through the top-level object's `Session` property.

Session Object Properties and Methods

Properties	Methods
Connected	Connect
LocalAddress	Disconnect
RemoteAddress	Log
RemoteHostName	
RemoteHostPort	
SSHPrivateKeyFile	
User	
Pass	
LogFileName	

Connected

Description

Returns a Boolean value indicating whether the current session is connected or not.

Syntax

object.Connected

Remarks

Boolean read-only property.

LocalAddress

Description

Returns the IP address of the local machine in the form of a string.

Syntax

object.localAddress

Remarks

LocalAddress is a read-only string property. The LocalAddress property should only be accessed if the session is connected. Attempting to access LocalAddress while not connected is an error.

RemoteAddress

Description

Returns the IP address of the remote host in the form of a string.

Syntax

object.RemoteAddress

Remarks

RemoteAddress is a read-only string property. The RemoteAddress property should only be accessed if the session is connected. Attempting to access RemoteAddress while not connected is an error.

RemoteHostName

Description

Returns the remote host name in the form of a string. This is the contents of the Profile Manager field “Host”.

Syntax

object.RemoteHostName

Remarks

RemoteHostName is a read-only string property.

RemoteHostPort

Description

Returns the remote host port address in the form of a short integer. This is the contents of the Profile Manager field “Port”.

Syntax

object.RemoteHostPort

Remarks

RemoteHostPort is a read-only string property.

SSHPrivateKeyFile

Description

Returns the filename of the SSH private key file in the form of a string.

Syntax

object.SSHPrivateKeyFile

Remarks

SSHPrivateKeyFile is a read-only string property.

User

Description

Set/Get the login userid. Can be used in protocols such as SSH to provide userid. A set will override the userid set in the Site Profile Manager (if any).

Syntax

object.User

Remarks

Set/Get string property.

Pass

Description

Set/Get the login user password. Can be used in protocols such as SSH to provide users password. A set will override the password set in the Site Profile Manager (if any).

Syntax

object.Pass

Remarks

Set/Get string property.

LogFileName

Description

Returns or sets the name of the current log file.

Syntax

object.LogFileName [= *filename*]

Remarks

If filename is invalid a runtime error is generated. See also: Log

Connect

Description

Connects to a session. Returns TRUE if the connection attempt was started. Returns FALSE if a session is already connected, if the specified host cannot be found in the Site Profile Manager or the connection cannot be started. A return of TRUE only indicates that the connection was started, not that a connection to the host was made.

Syntax

object.Connect *arg*

Remarks

The Connect method takes a string parameter that specifies how a connection is to be made.

Examples:

```
snt.Connect("Active")
```

Disconnect

Description

Disconnects the current session.

Syntax

object.Disconnect

Remarks

If the current session is not connected, disconnect does nothing.

Log

Description

Enables or disables logging.

Syntax

object.Log(start[,append)

Remarks

Starts or stops logging depending on the Boolean state of the ‘start’ parameter. When logging is being started the optional Boolean ‘append’ parameter may be set to True to open the log file for appending. If append is not specified, the default is False.

Screen Control

Description

The Screen properties and methods provides access to SecureNetTerm's terminal screen.

Screen Object Properties and Methods

Properties	Methods
Autowrap	Clear
Current Column	Get
Current Row	Print
Columns	Send
Rows	QuickButton
Synchronous	WaitForString
	WaitForStrings
	CopyScreenToClipboard
	CopyScrollToClipboard

Autowrap

Description

Returns or sets the line control Autowrap setting of the screen.

Syntax

object.Autowrap [= True | False]

Remarks

Autowrap mode controls what happens when text printed in the SNetTerm window reaches the right-hand edge of the window.

With autowrap 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.

CurrentColumn

Description

Returns the current column of the cursor.

Syntax

object.CurrentColumn

Remarks

Read-only numeric property. The first column is 1. An error will be returned if there is no connection open.

CurrentRow

Description

Returns the current row of the cursor.

Syntax

object.CurrentRow

Remarks

Read-only numeric property. The first row is 1. An error will be returned if there is no connection open.

Columns

Description

Returns the current number of columns.

Syntax

object.Columns

Remarks

Read-only numeric property.

Rows

Description

Returns the current number of rows

Syntax

object.Rows

Remarks

Read-only numeric property.

Synchronous

Description

Returns or sets the Synchronous setting of the screen.

Syntax

object.Synchronous [= True | False]

Remarks

If Synchronous is False, then under certain circumstances a script can miss data sent by the server that it is expecting to see. Synchronous is set to False by default, and should only be set to true in circumstances when a possibility of a script missing data exists. Scripts should then return the Synchronous state to False.

In general, Synchronous should only be set to True when you are performing an activity that would result in a **continuous** flow of data from the server, but want to be able to wait for strings within that continuous flow of data. A good example of this is the GetExternalData.vbs script in which the contents of a file is sent from the host, and the script desires to capture each line. Since most applications. such as SecureNetTerm, receives this data in a real time manner, and more than likely the entire contents of the file will be received in one network buffer, trying to capture individual strings within that buffer would not normally be possible. Setting synchronous to True instructs SecureNetTerm to add additional internal buffering to "sink" the host, network and SecureNetTerm logic to the script.

It is imperative to set Synchronous back to False after any of these type of operations. If for any reason, Synchronous to not set back to False, an Edit-Terminal Reset will have to be issued to resume normal operations.

Clear

Description

Clears the screen.

Syntax

object.Clear

Remarks

None.

Get

Description

Returns a string of characters read for a portion of the screen.

Syntax

object.Get(row1, col1, row2, col2)

Remarks

Returns a string containing the characters on the screen rectangle defined by the numeric values row1,col1 (upper-left) and row2,col2 (lower-right). The rows/columns should be specified starting at 1. If the current screen size is 24 rows of 80 columns each, then to get the entire screen

```
snt.Get(1,1,24,80)
```

Rows/columns are padded with spaces to the maximum number of rows/columns, therefore the Get shown above will always return 24 rows of 80 columns, regardless of the number of lines currently displayed on the screen.

End of line characters (\r, \n, \r\n) are not returned.

The current screen rows/columns can be obtained with the Rows/Columns properties.

Print

Description

Prints the screen

Syntax

object.Print

Remarks

If no printer is defined on your machine, an error will be returned.

Send

Description

Sends a string of characters to the connected host.

Syntax

object.Send *string*

Remarks

Attempting to send a string while no connection is open returns an error.

QuickButton

Description

Executes a SecureNetTerm QuickButton style command. Refer to the QuickButton documentation for details.

Syntax

object.QuickButton *string*

Examples:

```
snt.QuickButton "mc^M"
snt.QuickButton "@notepad.exe\myfile.txt"
snt.QuickButton "~&Certificates…"
snt.QuickButton "http://www.securenetterm.com"
```

WaitForString

Description

Wait for a string.

Syntax

object.WaitForString *string* [, *timeout*]

Remarks

Wait for the string to appear in the input. The timeout (seconds) parameter is optional, and will default to 30 seconds if not specified. When the string is detected in the input WaitForString() returns True. If a timeout occurs the function returns False. An error will be returned if there is no connection open.

Example:

```
If snt.WaitForString("ogin:", 10) <> True Then
MsgBox "Failed to detect login!"
Exit Sub
End If
```

WaitForStrings

Description

Wait for one of several strings to appear in the input.

Syntax

object.WaitForStrings Array [, *timeout*]

Remarks

Waits for one of the strings contained in Array. When one of the argument strings is matched in the Array, WaitForStrings() returns the argument index of the string that was found (the index of the first string within the Array is 1). If the optional timeout parameter is specified and a timeout occurs before any of the strings are found, WaitForStrings() returns 0. In the absence of a timeout, a default value of 30 seconds will be used. An error will be returned if there is no connection open.

Example:

```
Dim result,waitForStrs
WaitStrs = Array("string1", "string2", "string3")
result = snt.WaitForStrings(waitForStrs, 10)
MsgBox result
If result = 3 Then
MsgBox "Got string3!"
End If
If result = 0 Then
   MsgBox "Timed out!"
End If
```

CopyScreenToClipboard

Description

Copy current screen text to the clipboard.

Syntax

object.CopyScreenToClipboard

CopyScrollToClipboard

Description

Copy the scroll buffer text to the clipboard.

Syntax

object.CopyScrollToClipboard

Dialogs

Description

The Dialog related methods provides access to simple user-interface features provided by SecureNetTerm.

Dialog Object Methods

Methods
Prompt
MessageBox
FontDialog

Prompt

Description

Prompt the user to enter a string.

Syntax

object.Prompt(*message* [, *title [, default [, isPassword]]*])

Remarks

The Prompt function displays a simple dialog that has message and an edit field for the user to enter a string. The message parameter is an informational string displayed in the prompt dialog. Optionally the title of the prompt dialog may be set by passing a title string. By default the edit field is empty, but the initial contents of the edit field may be set with the optional default string. Finally, if the text entered in the edit field is to be obscured as it is entered (such as when entering a password) then the Boolean 'isPassword' field should be set to True.
If the user clicks OK, Prompt returns the entered string.

Example:

```
Dim pass
pass = snt.Prompt("Enter your password:", "Logon Script", "", True)
```

MessageBox

Description

Display a message

Syntax

object.MessageBox(*message* [, *title* [, *options*]])

Remarks

The MessageBox function displays a message string to the user. The optional title string sets the title or caption of the MessageBox. The buttons that appear on the MessageBox can be configured by passing a combination of numeric values in the optional 'options' parameter. By default, the MessageBox will display the message string with an **OK** button. However, many possibilities exist for displaying message boxes with different icons, and buttons.
The MessageBox function returns a numeric value that can be used to identify which button was clicked.

The following code sample defines the constants that can be combined to form the 'options' parameter as well as the possible numeric return values:

```
' options parameter
Const ICON_STOP = 16              ' display the ERROR/STOP icon.
Const ICON_QUESTION = 32            ' display the '?' icon
Const ICON_WARN = 48               ' display a '!' icon.
Const ICON_INFO= 64               ' displays "info" icon.

Const BUTTON_OK = 0                ' OK button only
Const BUTTON_CANCEL = 1             ' OK and Cancel buttons
Const BUTTON_ABORTRETRYIGNORE = 2    ' Abort, Retry,Ignore buttons
Const BUTTON_YESNOCANCEL = 3         ' Yes, No, and Cancel buttons
Const BUTTON_YESNO = 4              ' Yes and No buttons
Const BUTTON_RETRYCANCEL = 5         ' Retry and Cancel buttons

Const DEFBUTTON1 = 0       ' First button is default
Const DEFBUTTON2 = 256      ' Second button is default
Const DEFBUTTON3 = 512      ' Third button is default

' Possible MessageBox() return values
Const IDOK = 1             ' OK button clicked
Const IDCANCEL = 2          ' Cancel button clicked
Const IDABORT = 3           ' Abort button clicked
Const IDRETRY = 4           ' Retry button clicked
Const IDIGNORE = 5          ' Ignore button clicked
Const IDYES = 6             ' Yes button clicked
Const IDNO = 7             ' No button clicked

' Display a messagebox with Yes/No buttons.
' Make the 'No' button the default.
result = snt.MessageBox("Login Failed, Retry?", "Error", ICON_QUESTION Or
BUTTON_YESNO Or DEFBUTTON2 )
If result = IDNO Then
Exit Sub
End If
```

FontDialog

Description

Prompt the user to select a display font.

Syntax

object.FontDialog([, *title*])

Remarks

The FontDialog function displays a standard font dialog with an optional title requesting the user to select a new display font. If the user clicks OK, a 1 is returned, else 0.

Example:

```
Dim pass
Result = snt.FontDialog("Select Courier New, Western Script")
```

Zmodem File Transfers

Description

The FileTransfer related properties/methods provide script initiated file transfers.

FileTransfer Object Properties and Methods

Properties	Methods
UploadFolder	AddToZModemUploadList
DownloadFolder	ZmodemTransfer

ZModemTransfer

Description

File transfer using zmodem protocol.

Syntax

object.ZModemTransfer(Send,Wait,Command)

Remarks

The boolean Send is set to TRUE for sending files from the host to the workstation, otherwise it is FALSE. The boollean Wait is not currently implemented. The string Command is the UNIX host command used to start the transfer.

Example:

```
' Send a local file to the host
snt.AddToZmodemUploadList snt.DownloadFolder + "passwd"
ret = snt.ZModemTransfer(FALSE,FALSE,"rz" + Chr(10))
' Send a host file to the workstation
 ret = snt.ZModemTransfer(TRUE,FALSE,"sz /etc/passwd" + Chr(10))
```

UploadFolder

Description

Returns/Sets path for session upload folder.

Syntax

object.UploadFolder

Remarks

UploadFolder is a read/write property that returns/sets the path for the global upload folder.

Example:

```
MsgBox "Upload completed to: " & vblf & snt.UploadFolder
```

DownloadFolder

Description

Returns/Sets path for session download folder.

Syntax

object.DownloadFolder

Remarks

DownloadFolder is a read/write property that returns/sets the path for the global download folder. Note: Files downloaded are always placed in the global download folder.

Example:

```
MsgBox "Download completed to: " & vblf & snt.DownloadFolder
```

AddToZModemUploadList

Description

Places file on ZModem upload list.

Syntax

object.AddToZModemUploadList <filepath>

Remarks

AddToZModemUploadList places the specified file on a list of files that will be uploaded during the next ZModem upload. Once one or more files have been added to the upload list, a ZModem upload can be initiated by the script sending the appropriate command to the remote system.

Errors:

If the path provided to AddToZModemUploadList is not a valid file, a script error is generated and the following message is displayed:

```
"snt.AddToZModemUploadList: <filepath> does not exist."
```

Example:

```
snt.AddToZModemUploadList "c:\temp\File1.txt"
snt.AddToZModemUploadList "c:\temp\File2.txt"

' Start the upload of the two files...
ret = snt.ZModemTransfer(FALSE,FALSE,"rz" + Chr(10))
```

SFTP File Transfers

Description

The SFTP FileTransfer methods provide script initiated file transfers using the **SSH-2** secure protocol. An open SSH-2 interactive session must already have been established with the host. All of the SFTP script commands operate over a fully encrypted data channel.

Note that all file/path names are internally converted to/from ASCII when transferring them to/from the host. That is, all file/path names are in ASCII format when presented to the host.

The internal SFTP file object supports the SFTP_Stat and SFTP_ReadFileTree commands, but these two commands are not currently implemented. Refer to the initial specifications below.

FileTransfer Object Methods

Methods
SFTP_OpenConnection
SFTP_CloseConnection
SFTP_GetFile
SFTP_PutFile
SFTP_ChMod
SFTP_DeleteFile
SFTP_RenameFile
SFTP_CreateDirectory
SFTP_RemoveDirectory
SFTP_SetCurrentDirectory
SFTP_GetCurrentDirectory

SFTP_OpenConnection

Description

Open the SFTP file transfer channel.

Syntax

object.SFTP_OpenConnection

Remarks

Returns TRUE is successful, else FALSE. The file transfer channel must be opened prior to performing any other SFTP file transfer command. This command starts the host SFTP server.

SFTP_CloseConnection

Description

Close the SFTP file transfer channel.

Syntax

object.STP_CloseConnection

Remarks

Returns TRUE is successful, else FALSE. This command releases the SFTP object memory, closes the host SFTP channel and releases the host SFTP server.

SFTP_GetFile

Description

Transfers the specified host file to the workstation.

Syntax

object.SFTP_GetFile(remotefile,localfile [, *options*]])

Remarks

If the remotefile does not have a path specified, the default directory is used. The default directory is initially set to the users home directory. If the SFTP_SetCurrentDirectory method has been called, then the directory specified in that method becomes the default directory.

If the localfile does not have a path specified, the "Global Download" directory will be used. If the "Global Download" directory has not been specified, then the "\" directory will be used.

The optional "options" consist of 0 or 1, where 0 implies download in binary mode and 1 implies download in ASCII mode. If options is not specified, a binary transfer is used. If a file is transferred in ASCII mode, all UNIX style text end of line values (0x0a) is converted to Microsoft text end of line values (0x0a,0x0d).

Example:

```
ret = snt.SFTP_GetFile("myremotefile.txt","/temp/local.txt",1)
```

SFTP_PutFile

Description

Transfers the specified local workstation file to the host.

Syntax

object.SFTP_PutFile(localfile,remotefile [,options])

Remarks

If the remotefile does not have a path specified, the default directory is used. The default directory is initially set to the users home directory. If the SFTP_SetCurrentDirectory method has been called, then the directory specified in that method becomes the default directory.

If the localfile does not have a path specified, the "Global Upload" directory will be used. If the "Global Upload" directory has not been specified, then the "\" directory will be used.

The optional "options" consist of 0 or 1, where 0 implies download in binary mode and 1 implies download in ASCII mode. If options is not specified, a binary transfer is used. If a file is transferred in ASCII mode, all Microsoft style text end of line values (0x0a,0x0d) is converted to UNIX text end of line values (0x0a).

Example:

```
ret = snt.SFTP_GetFile("/temp/logo.gif","remotelogo.gif",0)
```

SFTP_ChMod

Description

Changes the specified file permissions.

Syntax

object.SFTP_ChMod(remotefile,command)

Remarks

The command is a numeric mode, composed of one to four octal digits (0-7), derived by adding up the bits with values 4, 2, and 1. Any omitted digits are assumed to be leading zeros. The first digit selects the set user ID (4) and set group ID (2) and sticky (1) attributes. The second digit selects permissions for the user who owns the file: read (4), write (2), and execute (1); the third selects permissions for other users in the file's group, with the same values; and the fourth for other users not in the file's group, with the same values.

Example:

```
ret = snt.SFTP_ChMod("mylogo.zip","644")
```

SFTP_DeleteFile

Description

Deletes the specified host file.

Syntax

object.SFTP_DeleteFile(filepath)

Remarks

Returns TRUE if successful or FALSE if an error occurred. If the remotefile does not have a path specified, the current directory is used.

Example:

```
ret = snt.SFTP_DeleteFile("File1.txt")
```

SFTP_RenameFile

Description

Renames the specified host file.

Syntax

object.SNT_RenameFile(oldname,newname)

Remarks

Returns TRUE if successful or FALSE if an error occurred. If the remotefile does not have a path specified, the current directory is used.

Example:

```
ret = snt.SFTP_RenameFile("oldfile.txt","newfile.txt")
```

SFTP_CreateDirectory

Description

Creates a new directory on the host.

Syntax

object.SNT_CreateDirectory("path")

Remarks

Returns TRUE if successful or FALSE if an error occurred. If the remotefile does not have a path specified, the current directory is used.

Example:

```
ret = snt.SFTP_CreateDirectory("UploadDirectory")
```

SFTP_RemoveDirectory

Description

Removes (deletes) a host directory.

Syntax

object.SNT_RemoveDirectory("path")

Remarks

Returns TRUE if successful or FALSE if an error occurred. If the remotefile does not have a path specified, the current directory is used.

Example:

```
ret = snt.SFTP_RemoveDirectory("UploadDirectory")
```

SFTP_SetCurrentDirectory

Description

Sets/Changes the base directory.

Syntax

object.SFTP_SetCurrentDirectory(path)

Remarks

Returns TRUE if successful, else false. Changes the host base directory to <path>. The ability to change to a directory not contained within the login "home' directory is dependent upon your host security rules.

Example:

```
ret = snt.SFTP_SetCurrentDirectory("/")
```

SFTP_GetCurrentDirectory

Description

Returns a string containing the current host directory.

Syntax

object.SFTP_GetCurrentDirectory

Remarks

The SFTP_OpenConnection command starts the host SFTP server and will set the initial directory to that established with the interactive session login. This will normally be the "home" directory of the userid the session was logged in with. All file/directory related commands will use this as the base directory, if a specific directory is not included within the command.

Example:

```
dir = snt.SFTP_GetCurrentDirectory
```

SFTP_Stat

Description

Returns a structure containing detailed information relating to a file/directory.

Syntax

object.SFTP_Stat("filename")

Remarks

Not currently implemented.

Internally, the SFTP_Stat command returns the following structure:

```
    Struct Attrib
        {
        u_int32_t flags;
        u_int64_t size;
        u_int32_t uid;
        u_int32_t gid;
        u_int32_t perm;
        u_int32_t atime;
        u_int32_t mtime;
        };
```

SFTP_ReadFileTree

Description

Returns a pointer to an array of SFTP_DIRENT stuctures.

Syntax

object.SFTP_ReadFileTree("path",flags)

Remarks

Not currently implemented.

Internally, the SFTP_ReadFileTree returns a SFTP_DIRENT ** to an array of SFTP_DIRENT stuctures.

```
struct SFTP_DIRENT
    {
    char *filename;
    char *longname;
    Attrib a;
    };
```

Events

SecureNetTerm contains advanced support for events. Events are a method by which SecureNetTerm can notify your script about something that has changed, such as cursor and mouse movement. When SecureNetTerm is controlled from another program, which supports scripting, events must be "mapped" from that program to the event notifier within SecureNetTerm. This is done with a special program supplied with SecureNetTerm, referred to as the SecureNetTerm EventMapper. The use of this program is triggered automatically within the script through the use of the scripting language support for creating objects. The VBScript example shown below demonstrates the creation of the SecureNetTerm EventMapper and how to map the SecureNetTerm events you desire to use in your script.

The example script maps all the events supported by SecureNetTerm. Note the "OnDisconnected" event. When this event is received, the script sets the global variable "Answer" to a value of 6. In the Do While loop toward the bottom of the example, this variable is checked. It the variable has a value of 6, the script will display a message dialog, asking if the script should be stopped.

Note that scripts which run within SecureNetTerm does not require the use of the EventMapper.

Event Name	**Input Parameters**	**Remarks**
OnConnected		Connected to host
OnDisconnected		Disconnected from host
OnKeyDown	ByVal type, ByVal wParam,ByVal lParam	Key was pressed
OnTokenFound	ByVal index	WaitForString token found
OnMouseLUp	ByVal x, ByVal y	Mouse left button up
OnMouseLDown	ByVal x, ByVal y	Mouse left button down
OnMouseMUp	ByVal x, ByVal y	Mouse middle button up
OnMouseMDown	ByVal x, ByVal y	Mouse middle button down
OnMouseRUp	ByVal x, ByVal y	Mouse right button up
OnMouseRDown	ByVal x, ByVal y	Mouse right button down
OnMouseLDblClk	ByVal x, ByVal y	Mouse left button double click
OnMouseMove	ByVal x, ByVal y	Mouse moved
OnCursorMove	ByVal x, ByVal y	Cursor moved

The OnKeyDown event is sent on WM_KEYDOWN, WM_SYSKEYDOWN and WM_CHAR notifications to the SecureNetTerm program. The WM_CHAR notification event is triggered when a KW_KEYDOWN message is translated by the TranslateMessage function within SecureNetTerm. The parameter "type" is WM_CHAR, WM_SYSKEYDOWN or WM_SYSKEYDOWN. The wParam and lParam parameters are the same values passed to SecureNetTerm from the OS when a keyboard key is pressed, and is documented in normal Microsoft documentation.

VBScript Example:

```
'# $language = "VBScript"
'# $interface = "1.0"

' Globals
dim snt
dim sink
dim Connected
dim KeyDown
dim Answer

Sub snt_OnMouseLUp(ByVal x, ByVal y)
        snt.StatusLine "Left Mouse Up at x=" & x & "  y=" & y
End Sub

Sub snt_OnMouseLDown(ByVal x, ByVal y)
        snt.StatusLine "Left Mouse Down at x=" & x & "  y=" & y
End Sub

Sub snt_OnMouseMUp(ByVal x, ByVal y)
        snt.StatusLine "Middle Mouse Up at x=" & x & "  y=" & y
End Sub

Sub snt_OnMouseMDown(ByVal x, ByVal y)
        snt.StatusLine "Middle Mouse Down at x=" & x & "  y=" & y
End Sub

Sub snt_OnMouseRUp(ByVal x, ByVal y)
        snt.StatusLine "Right Mouse Up at x=" & x & "  y=" & y
End Sub

Sub snt_OnMouseRDown(ByVal x, ByVal y)
        snt.StatusLine "Right Mouse Down at x=" & x & "  y=" & y
End Sub

Sub snt_OnMouseMove(ByVal x, ByVal y)
        snt.StatusLine "MouseMove at x=" & x & "  y=" & y
End Sub

Sub snt_OnMouseLDblClk(ByVal x, ByVal y)
        snt.StatusLine "Left Double click at x=" & x & "  y=" & y
End Sub

Sub snt_OnCursorMove(ByVal x, ByVal y)
        snt.StatusLine "Cursor at x=" & x & "  y=" & y
End Sub

Sub snt_OnConnected()
        Connected = True
End Sub

Sub snt_OnDisconnected()
        Connected = False
        Answer = snt.MessageBox("Quit SecureNetTerm application?",,4)
End Sub

Sub snt_OnKeyDown(ByVal type,ByVal wParam,ByVal lParam)
        KeyDown = True
End Sub

Sub snt_OnTokenFound(ByVal index)
```

```
End Sub


Sub StartSNT
  ' Request to handle errors ourselves so we can handle possible failure
  ' of GetObject() to connect to SecureNetTerm...
  '
  On Error Resume Next
  ' Gets an instance of SecureNetTerm if it is already running
  Set snt = GetObject("SecureNetTerm.Document")
  If TypeName(snt) <> "Document" Then
    ' It wasn't already running so start it running.
    Set snt = CreateObject("SecureNetTerm.Document")
  End If
End Sub

Sub AttachEvents
 Set sink = CreateObject("EventMapper.SecureNetTerm")
 sink.Init snt,"OnMouseMove",GetRef("snt_OnMouseMove")
 sink.Advise "OnMouseLUp",GetRef("snt_OnMouseLUp")
 sink.Advise "OnMouseLDown",GetRef("snt_OnMouseLDown")
 sink.Advise "OnMouseMUp",GetRef("snt_OnMouseMUp")
 sink.Advise "OnMouseMDown",GetRef("snt_OnMouseMDown")
 sink.Advise "OnMouseRUp",GetRef("snt_OnMouseRUp")
 sink.Advise "OnMouseRDown",GetRef("snt_OnMouseRDown")
 sink.Advise "OnMouseLDblClk",GetRef("snt_OnMouseLDblClk")
 sink.Advise "OnCursorMove",GetRef("snt_OnCursorMove")
 sink.Advise "OnConnected",GetRef("snt_OnConnected")
 sink.Advise "OnDisconnected",GetRef("snt_OnDisconnected")
 sink.Advise "OnKeyDown",GetRef("snt_OnKeyDown")
 sink.Advise "OnTokenFound",GetRef("snt_OnTokenFound")
End Sub

Sub Main()
 Dim x,y,pass
 StartSNT()
 AttachEvents()
 KeyDown = False
 Connected = False
 Answer = 0
 snt.Visible = True
 snt.WindowState = 1 'Normal (SW_SHOW)
 x = snt.GetViewWidth
 y = snt.GetViewHeight
 snt.StatusLine "Screen: x=" & x & "  y=" & y
 snt.Connect("Active")
'************************************************************
' Remove comment flag on the following if
' username/password processing is required.
' If snt.WaitForString("login:",15) <> TRUE Then
'    Exit Sub
' End if
' snt.Send "zkrr01" + Chr(10)
' If snt.WaitForString("Password:",15) <> TRUE Then
 '    Exit Sub
' End if
' pass = snt.Prompt("Password",,,TRUE)
' snt.Send pass + Chr(10)
'*****************************************************
 Do While snt.Visible
 ' Sleep(0) will just give up timeslice
 sink.Sleep(0)
 ' Note that Answer is set in the Disconnect event
 if Answer = 6 Then
  sink.Close
  snt.QuitApp
  Exit Sub
 End If
 Loop
End Sub

Main()
```

Advanced Support

Special Escape Sequences

SecureNetTerm contains several special escape definitions that are not a part of the published VT-XXX or ANSI standards. These have been requested by several of our clients to enhance the functionality of SecureNetTerm and provide a more flexible client interface for their UNIX programs. The following are the special escape codes:

^[[]**URL**^[[0*	Start/run the program to process the **URL**.
^[[]**COMMAND**^[[1*	Start/run the program specified by **COMMAND**.
^[[]**COMMAND**^[[2*	Define the International keyboard/video map to use.
^[[]**COMMAND**^[[3*	Define the keyboard definition template to use.
^[[]**COMMAND**^[[5*	Define and execute a QuickButton style command .
^[[]**DIRECTORY**^[[6*	Change the file transfer download directory to **DIRECTORY**.
^[[]**DIRECTORY**^[[7*	Change the file transfer upload directory to **DIRECTORY**.
^[[]**FILENAME**^[[8*	Define the file(s) to upload on the next upload request. **FILENAME** must contain a complete pathname. If **FILENAME** has a quote delimited string prior to the actual filename, the quotes will be stripped and the resulting text will be sent to the host. For example, **"rz" c:/work/myfile.txt** will define c:\work\myfile.txt as the file to upload and SecureNetTerm will send the string rz to the host, resulting in an automatic zmodem upload.
^[[]**COMMAND**^[[9*	Start/run the program specified by **COMMAND** and **wait** till it terminates. **COMMAND** can contain both the program to run and command line arguments.
^[[]**COMMAND**^[[10*	Check the status of the 1, 9, 12 and 13 special escape requests. If the operation was successful, an ASCII 1 will be send to the host, else an ASCII 0.
^[[]**TEXT**^[[11*	Place the string **TEXT** on the Windows clipboard.
^[[]**FILENAME**^[[12*	Start the program associated with the extension contained within **FILENAME**.
^[[]**DIRECTORY**^[[13*	Change the current working directory to **DIRECTORY**.

^[[]**User Information**^[[16*	Bring up the Setup Printer dialog panel. The optional **User Information** string is displayed in the StatusBar. An ASCII "0" will be returned if the user selects the cancel button, else an ASCII "1" will be returned. This escape sequence can be used to allow a user to select another printer/font/font size for a transparent print request.
^[[]**OPTION**^[[17*	This escape sequence will start the program associated with the file name extension of the next zmodem file downloaded. The variable **OPTION** can have a value of **WAIT**, **WAITASCII** or **NOWAIT**. If **WAIT** is specified, SNetTerm will wait for the started program to exit, and if the file has been changed, it will send the file back to the host using zmodem. If **WAITASCII** is specified, it will take the same action as **WAIT**, and in addition, will use the zmodem option to transfer the file back to the host in ascii mode. If **NOWAIT** is specified, no action will be taken after the program associated with the file has been started. If the variable **OPTION** has the value of **PRINT**, the next file downloaded by zmodem will be passed to the printer subsystem for printing using the options specified in the Global Settings-Printing dialog.
^[[]**PAGE**^[[18*	Bring up the Global Settings dialog. The optional **PAGE** defines which page to set active within the dialog. If **PAGE** is not provided, it will default to the first page (Terminal-Options). The dialog pages are numbered starting at 1 and are sequential to the end of the dialog (that is 2 would be the Printing page and 8 would be the Logo page). An ASCII "0" will be returned if the user selects the cancel button, else an ASCII "1" will be returned.
^[[=M**MESSAGE**^[[=S	Display **MESSAGE** in the StatusBar
^[[=m**TITLE**^[[=S	Set the SecureNetTerm window title to **TITLE.**

Where ^[is the ESC character (0x1b), **URL** is any valid URL and **COMMAND** is any valid DOS or Windows program. The maximum length of values **MESSAGE** and **TITLE** is 80 characters; **URL, COMMAND, DIRECTORY, FILENAME** and **TEXT** have a maximum length of 2047 characters. When a full path and filename is specified, use the forward slash instead of the backward slash. For example:

d:/user/files/myeditor.exe

The ability to start/run programs on the local workstation can create serious security problems. For this reason, a global flag controls whether SecureNetTerm will honor a host request to start/run local programs. The default is **not** to allow any host to start/run local programs. If SecureNetTerm detects such a request, a message will be displayed indicating a security violation. In order to enable the ability to start/run programs, the option ‘Allow Program Calls’ must be enabled.

The ‘Allow Program Calls’ option does not apply to operations relating to the browser (URL) or to the editor (such as the transparent printing/netedit requests).

SecureNetTerm has also implemented the Locator Input Model for ANSI Terminals (sixth revision). This model defines a method to control a device connected to a serial communication port, and the ability to control a pointing device such as a mouse.

The locator controller mode allows the host to communicate directly with the locator device without terminal intervention. When locator controller mode is set, all data received at the host port is transferred directly to the locator port without interpretation by SecureNetTerm. All data received from the locator device will be sent directly to the host. The serial port for the locator device must be defined in the Global Options-Terminal-Locator Controller dialog panel.

The DEC defined sequence (MC) CSI 7i turns on the locator controller mode (opens the serial port), and the sequence (MC) CSI 6i turns off the locator controller mode and closes the serial port.

In addition to support for the locator controller mode, SecureNetTerm has implemented the Locator Input Model as defined by DEC. The Locator Input Model allows for maximum flexibility and control of mouse events from the host system. Refer to the DEC specification for complete details and required escape sequences.

International Video/Keyboard Mapping

SecureNetTerm maintains an internal mapping table for both the video and keyboard which allows defining what gets displayed/processed for both. This mapping method was developed for previous products, and should only be used if the National Replacement Character method cannot be used.

The video table, composed of 256 entries, allows the mapping of incoming network data to match the language where the program is being run. In general, mapping should only be used with an ANSI style font such as Courier New and NetTerm Ansi.

The keyboard mapping option allows for mapping the 'normal' keys to match the language where the program is being run. These keys are the ones that cannot be mapped in the keyboard definition dialog panel. Mapping is done based upon the keyboard scan code.

In addition, you can also use a character based keyboard mapping method which allows you to map any of the 256 possible outgoing characters to any other character. This method required manual editing of the mapping table.

The first step in Video/Keyboard mapping is to copy the model country.ini file (located in the SecureNetTerm directory) to another file prefixed with the desired country name, such as norway.ini. Then define this mapping file for your host in the Site Manager. The example sample file, norway.ini, contains the following mappings:

Video	Keyboard Normal	Keyboard Shifted
0x7b to 0xe6	0x1a to 0x7d	0x1a to 0x5d
0x7c to 0xf8	0x27 to 0x7c	0x27 to 0x5c
0x7d to 0xe5	0x28 to 0x7b	0x28 to 0x5b
0x5b to 0xc6		
0x5c to 0xd8		
0x5d to 0xc5		

National Replacement Characters

While 8-bit connections allow the display of multinational characters and special symbols, 7-bit connections cannot because there are fewer characters to display. To allow multinational characters over a 7-bit connection, the National Replacement Character (NRC) sets are used to map special language characters.

The NRC sets includes support for the United Kingdom, Dutch, Finish, French, French Canadian, German, Italian, Norwegian, Danish, Portuguese, Spanish, Swedish and Swiss language character sets. All the language specific characters are supported except for the Dutch ligature IJ. The NRC support requires the use of a windows ANSI based character set such as NetTerm ANSI and Courier New. The use of the National Replacement Characters is enabled in the Advanced Host Settings-Extended Options dialog panel.

Hexadecimal	**23**	**40**	**5B**	**5C**	**5D**	**5E**	**5F**	**60**	**7B**	**7C**	**7D**	**7E**
US ASCII	#	@	[	\	]	^	-	`	{	|	}	~

United Kingdom	£	@	[	\	]	^	-	`	{	|	}	~
Dutch	£	¾	[	½	|	^	-	`	¨	f	¼	´
Finish	#	@	Ä	Ö	Å	Ü	-	é	ä	ö	å	ü
French	£	à	º	ç	§	^	-	`	é	ù	è	¨
French Canadian	#	à	â	ç	ê	î	-	ô	é	ù	è	û
German	#	§	Ä	Ö	Ü	^	-	`	ä	ö	ü	ß
Italian	£	§	º	ç	é	^	-	ù	à	ò	è	ì
Norwegian/Dutch	#	Ä	Æ	Ø	Å	Ü	-	ä	æ	ø	å	ü
Portuguese	#	@	Ã	Ç	Õ	^	-	`	ã	ç	õ	~
Spanish	£	§	¡	Ñ	¿	^	-	`	º	ñ	ç	~
Swedish	#	É	Ä	Ö	Å	Ü	-	é	ä	ö	å	ü
Swiss	ù	à	é	ç	ê	î	è	ô	ä	ö	ü	û

Servers

Configure an SSH Data Communications Server

In order to use your public key you must transfer the identity .pub file created by the Security Manager to the ~/.ssh2 folder on the SSH host. It is recommended that you follow the procedure below to create a copy of the identity .pub file in the ~/.ssh2 folder on the remote machine.

The procedure outlined here assumes that you have the same account on both the SSH server and the FTP server and that they share files. If this is not the case, contact your system administrator for instruction on setting up your public-key files on your SSH server.

To configure the SSH server to recognize your identity .pub file:

1. On your local machine, use a text editor to create an empty file named authorization.
2. Connect to the remote server using SSH and password authentication.
3. On the server, create a ~/.ssh2 folder if necessary.
4. Using drag-and-drop, transfer the authorization file from the local window to the ~/.ssh2 folder in the remote window.
5. Using drag-and-drop, transfer the identity .pub file to the ~/.ssh2 folder.
6. Now add the line Key <identity>.pub to the authorization file (replacing <identity> with the name of your identity file).

The following steps outline how to do this in SecureNetTerm.

a. Select the authorization file in the remote window.
b. Double click on the file.
c. Select Notepad or your favorite editor when the Windows open with dialog opens.
d. The file will be downloaded to your local computer and the editor will be opened to edit the file.
e. Add the line Key <identity>.pub to the file, save the change, and exit the editor.
f. When you exit the editor, transfer the file back to the host by selecting the Toolbar-Commands-SmartEdit Save File option.

The method described above uses only a single public key in the authorization file. It is possible to have more than one public key in the authorization file. To do this, repeat steps 5 and 6. The names of the public-key files must be unique.

Note that the authorization file is a text style file, so if you editor will not handle UNIX style text files properly, select the Toolbar-Commands-Transfer Type-Ascii option.

Configure an OpenSSH Server

In order to use your public key you must transfer the identity .pub file created by the Security Manager to the ~/.ssh folder on the SSH host. It is recommended that you follow the procedure below to create a copy of the identity .pub file in the ~/.ssh folder on the remote machine.

The procedure outlined here assumes that you have the same account on both the SSH server and the FTP server and that they share files. If this is not the case, contact your system administrator for instruction on setting up your public-key files for the SSH server.

To configure the OpenSSH server to recognize your identity .pub file:

1. Connect to the remote server using SSH and password authentication.
2. On the server, create a ~/.ssh folder if necessary.
3. Using drag-and-drop, transfer the identity .pub file to the ~/.ssh folder. Be sure to transfer the file in binary mode.
4. Use a terminal emulator to login to the SSH host, change to the ~/.ssh folder and issue the following command:

```
cat identity.pub >> authorized_keys
```

This will append the contents of the identity.pub file to the authorize_keys file. Be sure to use the filename you transferred, instead of the example identity.pub filename. You can also download or double click on the authorized_keys file, then use a UNIX style editor to insert the new public key. However you must know that your editor will not corrupt this file, since it can contain very large lines.

The method described above uses only a single public key in the authorization file. It is possible to have more than one public key in the authorization file. To do this, repeat step 3 and 4.

Caution: OpenSSH 3.0 uses the authorized_keys file instead of authorized_keys2 file. Currently, the older authorized_keys2 file will still be recognized but this may change in future releases.

SFTP

Secure SFTP

Secure SFTP adds secure file transfers to/from the host computer using the current SSH connection. This feature requires a SSH protocol 2 (SSH-2) connection, and the host must support the SFTP protocol. SFTP is activated from the SecureNetTerm "Start SecureFTP" toolbar icon.

Operation

The SFTP main window consists of three sections; the host file structure, the local workstation file structure and a transfer status window. Files may be transferred to/from the host by the normal Windows drag-drop operation, the right mouse popup window, or by dropping files from other sources, such as the Microsoft Windows explorer.

Files transferred can accomplished in binary, ASCII or AutoASCII mode. Binary mode simply means files are transferred as is, with no modifications by the host or workstation SFTP programs. ASCII transfers implies that each record within a file must be processed and end of record characters must be converted from the Microsoft standard to the UNIX standard, depending upon the transfer direction. AutoASCII mode implies that SFTP will look at each file extension to determine whether the file should be transferred in ASCII mode or binary mode. The criteria SFTP uses to determine the transfer type is under the control of table of file extensions which define which files should be considered as ASCII. This table can be modified in the Global Setup dialog.

UNIX files starting with a period (".") are often referred to as hidden files, and normally are not displayed within the host file structure. The Options menu item "Toggle hidden file display" can be used to allow for the display of the hidden files.

The transfer status window displays information related to transfers; items displayed in green or black are considered informational items, whereas items displayed in red indicate a failure of some type. The vast majority of all failures are associated with UNIX file permissions, such as, a normal user trying to change into a directory accessable only by a system administrator or attemping to download a file restricted from the person attemping the download.

SFTP is only available as a result of an interactive SSH connection, and all user rights of the interactive login userid is granted to the SFTP file transfer session.

Configuration

The default host and local directories can be specified in the SecureNetTerm Advanced Host Settings, SFTP dialog. If the host directory is not specified, SFTP will default to the users home directory. If a local directory is specified, SFTP will display that directory upon startup.

Transfer File Types

The Transfer File Types panel defines the type of data transfer, and the files to be treated as Ascii for the Auto Ascii file transfer type. Auto Ascii is commonly used to instruct SecureFTP to transfer all files as binary, unless a file is encountered that has a file extension listed within the Auto Ascii file extension definition.

Ascii (text) files are handled differently on UNIX systems then they are on Microsoft based systems. The difference is in the end of line characters and in most cases, a conversion must take place when transferring ASCII text files from/to UNIX.

SecureKeyAgent

Key Agent

SecureKeyAgent is a windows based program to manage private keys used for SSH public key authentication. SecureKeyAgent supports disk based RSA and DSA private keys, and private keys contained within the Microsoft certificate store.

Private keys contained within the Microsoft certificate store can be located on Smart Cards, USB tokens and any other media supported by the Microsoft CryptoAPI.

SecureKeyAgent is normally started by the user when the workstation is first started. At that point, any private keys that have been defined to SecureKeyAgent will be processed. If a private key requires a passphrase for access, a dialog will be displayed prompting the user for a passphrase for that key.

Once started, SecureNetTerm runs silently in the background, with its icon placed within the taskbar system tray. Authentication requests will be forwarded to SecureKeyAgent from both SecureFTP and SecureNetTerm. The agent will then sign the authentication request and return the result to the respective requestor.

Private keys located on Smart Cards, USB tokens or the Microsoft store are accessed by the CryptoAPI and are not stored within the key agent itself, resulting in maximum protection of the private key.

Operation

SecureKeyAgent is placed in the taskbar system tray when started. User interaction with the agent is initiated with a right-mouse click on its icon. The menu offers the ability to display the keys currently known to the agent and the ability to add and delete keys.

Whenever the agent is busy with an authentication request, the icon will be changed to a red circle with an "X" indicating is it processing a request. Requests are normally processed in a few milliseconds, but if Smart Cards or tokens are involved, several seconds may be required, depending upon the device access speed.

Private keys can be defined to SecureKeyAgent with the "Create Agent Key List" menu item. When selected, a dialog will be presented to the user. The top half of the dialog handles disk based private key files; the bottom half handles private keys contained within the Microsoft certificate store. Keys can be selected for SSH-1, SSH-2 and SSH-2 X509 certificate authentication requests. The SSH-2 X509 certificate authentication (SSH2-X509) option should only be selected for SSH clients that support X509 certificate authentication.

The menu item "SecureKeyAgent" displays a dialog, which lists the keys currently known to the agent. The dialog also provides the ability to export the public key and or certificate for uploading to a host. The last column of the key list can be expanded to display the fingerprint of the public key.

Authentication requests can also be forwarded from the host SSH server to the key agent. Thus authentication data need not be stored on any other machine, and authentication pass phrases never go over the network.

Security

Access to the agent is restricted to the current user, or programs running under the current user windows login, if the workstation has security support (Windows NT, 2000 and XP). Access to the agent is through interprocess communications, not by network communications channels. Passphrases are only requested once, at agent startup.

Private keys stored on Smart Cards, USB tokens and other media are only accessed through the Microsoft CryptoAPI. They are never removed or copied from the device, providing maximum protection of the private key.

Control Information

Control information in maintained in the SecureCommon.ini file.

Supported Key Formats

SecureKeyAgent supports the private key formats of OpenSSH, SSH Data Communications, Putty and those defined by the Microsoft CryptoAPI, including all Smart Cards, USB tokens and other media support by that API.

Certificate Wizard

Cryptographic Service Provider

The “Cryptographic Service Provider” panel allows the selection of the service that will maintain your certificate. When a certificate and corresponding private key is created, it will be controlled by the service provider. The most common is the MicroSoft Enhanced Cryptographic Provider v1.0.

The validity months field specifies the number of months the certificate is valid for. The key size field is the size of the public/private key expressed in “bits”. The larger the value, the more secure the keys will be. The most common value is 1024. If you desire to be able to export your private key, select the “Exportable” check box. Note that some Cryptographic providers will not allow the private key to be exported. If the option “Copy certificate to device” is selected, the Certificate Wizard will attempt to write a copy of the certificate itself to the device. This option would only apply to Smart Cards and USB tokens, which support this ability.

Issued To

The “Issued To” panel allows you to enter all the data that you want to appear in the certificate issued to field. If you are creating a host certificate, all that is required is the Name/Host field, which would normally contain the fully qualified host name.

Enhanced Usage

The “Enhanced Usage” panel provides for enhanced/extended certificate options. Certificates used by SecureNetTerm and SecureFTP would normally have a key usage of “Exchange” and a certificate usage of “Client Authentication”.

The “Certificate Friendly Name” field, if supplied, will be placed in the Microsoft Certificate Store and does not appear within the certificate itself.

The “Create, Self Sign” and “Create, CA Sign” options specifies how the wizard will sign your certificate. If the CA sign option is selected, then the CA certificate/key to be used to sign the certificate will be requested. This certificate must be available from your workstation.

Acknowledgements

Icons

The SecureNetTerm icon was designed by Achim Vedam, vedam@a-vedam.de

Custom icons were designed by FOOOD Icons

SRP

```
 *
 * 1. Any software that incorporates the SRP authentication technology
 *    must display the following acknowledgment:
 *    "This product uses the 'Secure Remote Password' cryptographic
 *     authentication system developed by Tom Wu (tjw@CS.Stanford.EDU)."
 *
 * 2. Any software that incorporates all or part of the SRP distribution
 *    itself must also display the following acknowledgment:
 *    "This product includes software developed by Tom Wu and Eugene
 *     Jhong for the SRP Distribution (http://srp.stanford.edu/srp/)."
 *
 * 3. Redistributions in source or binary form must retain an intact copy
 *    of this copyright notice and list of conditions.
 */
```

OpenSSH

```
/*
 * Author: Tatu Ylonen <ylo@cs.hut.fi>
 * Copyright (c) 1995 Tatu Ylonen <ylo@cs.hut.fi>, Espoo, Finland
 *                    All rights reserved
 *
 * As far as I am concerned, the code I have written for this software
 * can be used freely for any purpose.  Any derived versions of this
 * software must be clearly marked as such, and if the derived work is
 * incompatible with the protocol description in the RFC file, it must be
 * called by a name other than "ssh" or "Secure Shell".
 *
 *
 * Copyright (c) 2000 Markus Friedl.  All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */
```

OpenSSH-X509 Certificate Authentication

```
/*
 * Copyright (c) 2004 Roumen Petrov.  All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */
```

OpenSSL

```
LICENSE ISSUES
  ==============

  The OpenSSL toolkit stays under a dual license, i.e. both the conditions of
  the OpenSSL License and the original SSLeay license apply to the toolkit.
  See below for the actual license texts. Actually both licenses are BSD-style
  Open Source licenses. In case of any license issues related to OpenSSL
  please contact openssl-core@openssl.org.

  OpenSSL License
  ---------------

/* ====================================================================
 * Copyright (c) 1998-2000 The OpenSSL Project.  All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in
*    the documentation and/or other materials provided with the
*    distribution.
*
* 3. All advertising materials mentioning features or use of this
*    software must display the following acknowledgment:
*    "This product includes software developed by the OpenSSL Project
*    for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
*
* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
*    endorse or promote products derived from this software without
*    prior written permission. For written permission, please contact
*    openssl-core@openssl.org.
*
* 5. Products derived from this software may not be called "OpenSSL"
*    nor may "OpenSSL" appear in their names without prior written
*    permission of the OpenSSL Project.
*
* 6. Redistributions of any form whatsoever must retain the following
*    acknowledgment:
*    "This product includes software developed by the OpenSSL Project
*    for use in the OpenSSL Toolkit (http://www.openssl.org/)"
*
* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
* EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
* OF THE POSSIBILITY OF SUCH DAMAGE.
* ====================================================================
*
* This product includes cryptographic software written by Eric Young
* (eay@cryptsoft.com).  This product includes software written by Tim
* Hudson (tjh@cryptsoft.com).
*
*/
```

Original SSLeay License

```
/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscape’s SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to.  The following conditions
* apply to all code found in this distribution, be it the RC4, RSA,
* lhash, DES, etc., code; not just the SSL code.  The SSL documentation
* included with this distribution is covered by the same copyright terms
* except that the holder is Tim Hudson (tjh@cryptsoft.com).
*
* Copyright remains Eric Young's, and as such any Copyright notices in
* the code are not to be removed.
* If this package is used in a product, Eric Young should be given attribution
* as the author of the parts of the library used.
* This can be in the form of a textual message at program startup or
* in documentation (online or textual) provided with the package.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
*    must display the following acknowledgement:
*    "This product includes cryptographic software written by
*     Eric Young (eay@cryptsoft.com)"
*    The word 'cryptographic' can be left out if the routines from the library
*    being used are not cryptographic related :-).
* 4. If you include any Windows specific code (or a derivative thereof) from
*    the apps directory (application code) you must include an acknowledgement:
*    "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
*
* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed.  i.e. this code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/
```

Kerberos

Copyright (C) 1985-1999 by the Massachusetts Institute of Technology.

All rights reserved.

Export of this software from the United States of America may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

WITHIN THAT CONSTRAINT, permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of M.I.T. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Furthermore if you modify this software you must label your software as modified software and not distribute it in such a fashion that it might be confused with the original MIT software. M.I.T. makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

THIS SOFTWARE IS PROVIDED ``AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Individual source code files are copyright MIT, Cygnus Support, OpenVision, Oracle, Sun Soft, FundsXpress, and others.

Project Athena, Athena, Athena MUSE, Discuss, Hesiod, Kerberos, Moira, and Zephyr are trademarks of the Massachusetts Institute of Technology (MIT). No commercial use of these trademarks may be made without prior written permission of MIT.

"Commercial use" means use of a name in a product or other for-profit manner. It does NOT prevent a commercial firm from referring to the MIT trademarks in order to convey information (although in doing so, recognition of their trademark status should be given).

The following copyright and permission notice applies to the OpenVision Kerberos Administration system located in kadmin/create, kadmin/dbutil, kadmin/passwd, kadmin/server, lib/kadm5, and portions of lib/rpc:

Copyright, OpenVision Technologies, Inc., 1996, All Rights Reserved

WARNING: Retrieving the OpenVision Kerberos Administration system source code, as described below, indicates your acceptance of the following terms. If you do not agree to the following terms, do not retrieve the OpenVision Kerberos administration system.

You may freely use and distribute the Source Code and Object Code compiled from it, with or without modification, but this Source Code is provided to you "AS IS" EXCLUSIVE OF ANY WARRANTY, INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, OR ANY OTHER WARRANTY, WHETHER

EXPRESS OR IMPLIED. IN NO EVENT WILL OPENVISION HAVE ANY LIABILITY FOR ANY LOST PROFITS, LOSS OF DATA OR COSTS OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, INCLUDING, WITHOUT LIMITATION, THOSE RESULTING FROM THE USE OF THE SOURCE CODE, OR THE FAILURE OF THE SOURCE CODE TO PERFORM, OR FOR ANY OTHER REASON.

OpenVision retains all copyrights in the donated Source Code. OpenVision also retains copyright to derivative works of the Source Code, whether created by OpenVision or by a third party. The OpenVision copyright notice must be preserved if derivative works are made based on the donated Source Code.

OpenVision Technologies, Inc. has donated this Kerberos Administration system to MIT for inclusion in the standard Kerberos 5 distribution. This donation underscores our commitment to continuing Kerberos technology development and our gratitude for the valuable work which has been performed by MIT and the Kerberos community.

Scintilla

License for Scintilla and SciTE

Copyright 1998-2003 by Neil Hodgson <neilh@scintilla.org>

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

NEIL HODGSON DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL NEIL HODGSON BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.