NAURTECH CORPORATION

SSH Addon

for Version 5.3

Naurtech CETerm SSH Addon

for Windows CE Devices

CETerm | CEVT220

draft

Copyright Notice

Trademarks

Software Version

This document is for version 5.3 of Naurtech CETerm or CEVT220 Terminal Emulation and Data Collection Web Browser smart clients.

Table of Contents

Preface

All of us at Naurtech Corporation constantly strive to deliver the highest quality products and services to our customers. We are always looking for ways to improve our solutions. If you have comments or suggestions, please direct these to:

Naurtech Corporation
e-mail: support@naurtech.com
Phone: +1 (425) 837.0800

Assumptions

This manual assumes you have working knowledge of:

- Microsoft Windows user interface metaphor and terminology.
- Stylus based touch screen navigation terminology.
- Basic operations and requirements of the host applications you want to access with the Naurtech smart client.

Conventions used in this Manual

This manual uses the following typographical conventions:

- All user actions and interactions with the application are in bold, as in **`[Session][Configure]`**

- Any precautionary notes or tips are presented as follows

  Tip: Text associated with a specific tip

- ☞ represents new version specific information

Additional Documentation

The Naurtech CETerm SSH Addon is an optional feature of Naurtech terminal emulation Smart Clients. Please refer to the User's Manual for these Smart

Clients for detailed installation and configuration information. The User's Manual may be downloaded from the "Support" section of our website. You may also want to refer to the CETerm Scripting Guide for additional features to enhance the product usability.

Online Knowledgebase

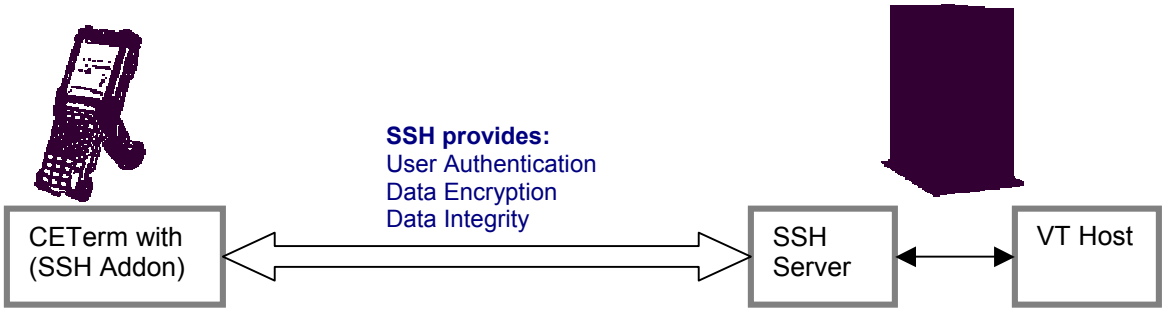
Although we continually strive to keep this manual up to date, you may find our online support knowledgebase useful for the latest issues, troubleshooting tips and updates. You can access the support knowledgebase from our website at:

www.naurtech.com → Support → Knowledgebase

1.0 Introduction

The CETerm SSH Addon is a tightly integrated implementation of Secure Shell protocols within CETerm for communication between the handheld terminal and a VT host. This feature provides a secure way to transmit data utilizing strong encryption and authentication to ensure confidentiality, authenticity and integrity of exchanged data.

SSH, or Secure Shell, is a popular, powerful, software-based approach to network security. Before data is sent by a computer over a network it is automatically encrypted (scrambled) by SSH. The data is automatically decrypted (unscrambled) when it reaches the intended recipient.
The result is transparent encryption: users can work normally, unaware that their communications are safely encrypted on the network. In addition, SSH uses modern, secure encryption algorithms and can be found within mission-critical applications at major corporations.



The SSH security capability within CETerm can be used to protect data in transit against the following security threats:

Password Exposure:
Eliminates the risk of password exposure in transmission of data between the handheld terminal and your VT host. When configured for SSH, CETerm sends passwords over the network in encyrpted format, making it impossible for outsiders to "sniff" the passwords.

Man in the middle attack:
A man-in-the-middle attack consists of attacker residing between the client and server and intercepting or modifying communications. With CETerm SSH, both client and server can authenticate and make cryptographic integrity checks to ensure that the transferred data has not be modified.

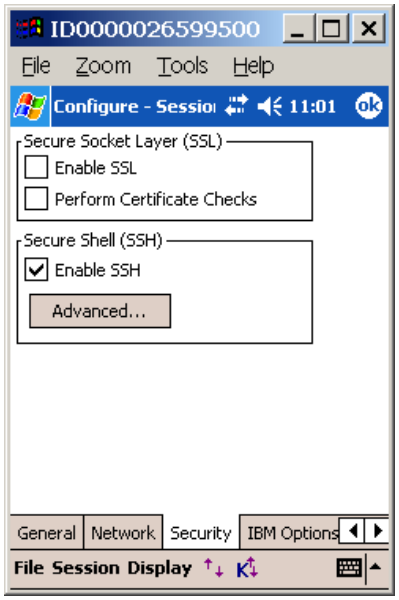
Data Eavesdropping:
With CETerm SSH, you prevent eavesdropping of confidential data, as it is encrypted while it travels over the networks. It also ensures that only the legitimate recipients can access the transmitted data.

1.1 Configuration

SECURITY

All SSH configuration settings within CETerm are under
[Session] -> [Configure] ->
[Connection] -> [Advanced] -> [Security]

Most SSH settings are specific to each session.



Enable SSH

This option enables the SSH protocol for this CETerm session. The default is Off.

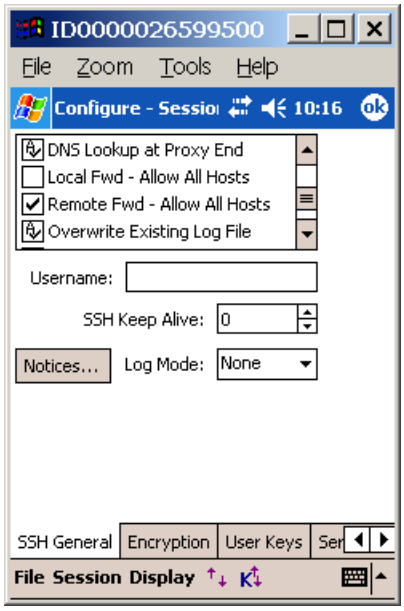
Advanced

This button allows configuration of more advanced settings specific to the SSH protocol. By default, these settings will enable connections to most SSH servers. It is recommended that you use the default settings, unless you understand implications of the various settings.

SSH GENERAL

This tab is used to configure general settings of the SSH protocol. The list contains check boxes that may be individually selected for the desired option. Please note that some items, whenever applicable, have three selectable states: On, Off and Auto-Sense (Check with “A”). Tapping these options will cycle through the three states.

If checked (On), the option is selected. If unchecked (Off), the option is not selected. If Auto-Sense, CETerm will determine and use the most appropriate option setting.



Enable SSH

This option enables SSH for the current session. The option is identical to the value on the previous Security tab and is repeated here for convenience.

Prefer SSH-1 Protocol

Enable this option if you would like to connect to your VT host server using SSH protocol version 1. The default setting is Off.

Prefer SSH-2 Protocol

Enable this option if you would like to connect to your VT host using SSH protocol version 2. The default setting is On.

NOTE: If both SSH-1 and SSH-2 options are off, CETerm will first try to connect using SSH2 protocol and then fall back to SSH-1 protocol.

Auto OK New Host keys

Automatically accept the host keys presented by the server. Enabling this option will minimize the amount of interaction that is required on part of a user. The default setting is Off.

WARNING: Enabling this option reduces security because the user does not verify that the server being connected to is the intended destination. For greatest security the user should verify that the server credentials presented match the intended server.

Auto OK Changed Host keys

Automatically accept changed host key presented by the server. Enabling this option will minimize the amount of interaction that is required on part of a user. The default setting is Off.

WARNING-WARNING-WARNING: Enabling this option reduces security because the user does not verify that the server being connected to is the intended destination. A changed host key may indicate a “man-in-the-middle” attack, or it may be that the server administrator has simply changed the server host key. For greatest security the user should verify that the server credentials presented match the intended server.

Try Keyboard Authentication

This option is for SSH-2 protocol only. It is a flexible authentication method using an arbitrary sequence of requests and responses. This method is not only useful for challenge/response mechanisms such as S/Key, but it can also be used for asking the user for a new password when the old one has expired.
This option is On by default.

Try TIS Authentication SSH-1

TIS authentication is a simple challenge/response form of authentication available in SSH-1 protocol only. You might use it if you were using S/Key one-time passwords, or if you had a physical security token that generated responses to authentication challenges.

With this option enabled, CETerm will attempt this authentication if the server is willing to try them. You will be presented with a challenge string (which will be

different every time) and must supply the correct response in order to log in. If your server supports this, you should talk to your system administrator about precisely what form these challenges and responses take. This option is Off by default.

Auth Username Changeable

The SSH-2 protocol allows change of username during authentication, but does not make it mandatory for SSH-2 servers to accept them. In particular, OpenSSH does not accept a change of username; once you have sent one username, it will reject attempts to try to authenticate as another user.

Enable this option if your server accepts changes to username in its authentication process. This option is Off by default.

Skip User Authentication

When enabled, CETerm will not negotiate user authentication with the SSH server. In most cases this will prevent a connection. This option is Off by default.

Cache Decrypted User Keys

When enabled, CETerm will retain a private copy of any user keys that have been unlocked with a user passphrase. Subsequent requests for the key will be served automatically and will not require the user to re-enter the passphrase. CETerm does not retain the passphrase. The cached key may be used with multiple sessions and will be erased when CETerm exits or if this option is changed to Off. This option is global and is common to all sessions. The default value is Off.

Re-Key Every 60 Minutes

A shared session key is used by the encryption protocol. If used too long, the session key may be subject to attack and expose the SSH connection. Although such an attack is unlikely, it is wise to re-exchange the key every so often. This can be initiated either by the client or the server. Enabling this option will trigger CETerm to exchange a new key with the server every 60 minutes. This option is On by default.

Note: While this renegotiation is taking place, no data can pass through the SSH connection, so the session may appear to ‘freeze’ momentarily. This is a short period when the key exchange is taking place.

Re-Key Every 1G of Traffic

A shared session key is used by the encryption protocol. If used for a large volume of traffic, the session key may be subject to attack and expose the SSH connection. Although such an attack is unlikely, it is wise to re-exchange the key after a significant amount of data. Enabling this option will trigger CETerm to exchange a new key with the server following a total data flow of 1 Gigabyte in either direction. This option is On by default.

Enable Compression

Enabling this setting will compress all data exchanged over the SSH connection. We do not recommend using compression for standard interactive sessions. By default, this setting is Off.

Enable Verbose messages

When enabled, the user is informed of major errors which affect the SSH connection via a popup dialog. Some additional information is presented within the emulation screen. This option is On by default.

Allow IPV4

Enable this option to allow the IPv4 Internet Protocol addressing scheme. The default is On .

Allow IPV6

Enable this option to allow the IPv6 Internet Protocol addressing scheme. The default is Off.

Note: If neither IPv4 or IPv6 options is selected, CETerm will use IPv4. If both are selected, CETerm will first attempt IPv6 and fall back to IPv4 if it is unsuccessful connecting with IPv6.

Use TCP No-Delay (Advanced)

Under normal operation, the TCP communication stack performs data packet batching. Enable this option forces the TCP stack to send immediately without batching data packets. This can result in excessive traffic of short packets. We recommend leaving this option Off. It is Off by default.

Send TCP Keep-Alives (Advanced)

Enables the TCP socket keep-alive option. This option is deprecated and should not be used. Use the SSH level keepalive to prevent session disconnection by a host. This option is Off by default.

No Pseudo-Terminal on Host (Advanced)

When connecting to a Unix system, most interactive shell sessions are run in a *pseudo-terminal*, which allows the VT host system to pretend it's talking to a real physical terminal device and allows the SSH server to catch all the data coming from that fake device and send it back to the client. Occasionally you might find you have a need to not run a session in a pseudo-terminal. Enable this option to prevent CETerm from running a pseudo terminal. The default is Off.

No Host Shell (Advanced)

Enabling this option will force CETerm to not run a shell or command after connecting to the remote host server / host. This option may be used only when using the SSH connection for port forwarding, and your user account on the server not having the ability to run a shell. This option is only applicable with SSH protocol version 2, since the SSH version 1 protocol assumes you will always want to run a shell. The default is Off.

Run Sub-System on Host (Advanced)

If enabled, attempts to run an SSH-2 subsystem on the host. By default, this option is Off

Try Proxy for Local Host

If using a proxy connection, this option enables the use of the proxy even for connections to localhost. By default, this option is Off

DNS Lookup at Proxy End

If Off, CETerm will perform DNS lookup on the handheld. If On, CETerm will perform DNS lookup on the proxy host. If Auto-Sense, CETerm will choose the DNS lookup location based on the proxy type. By default, this option is Auto-Sense.

Local Fwd – Allow All Hosts

If enabled, this option allows hosts other than the handheld to connect to local ports that are forwarded to the server. This may be useful for a peripheral device to connect to the server. By default, this option is Off

Remote Fwd – Allow All Hosts

If enabled, this option allows hosts other than the server to connect to remote ports that are forwarded to the handheld. By default, this option is Off

Overwrite Existing Log File

Enable this option if you want to automatically overwrite the existing log and start capturing a new log. If unchecked, log data will be appended to the end of the existing log. The default option is Auto-Sense, in which case the user is prompted when logging starts and the file exists.

Omit passwords from Log

When checked, password fields are removed from the log of transmitted packets. This includes any user responses to challenge-response authentication methods such as 'keyboard-interactive'. Note that this setting will only omit data that CETerm knows to be a password. If you start another login session within your CETerm SSH session, for instance, any password used will appear in the clear in the packet log. This option is Off by default.

Omit Session data from Log

When checked, all 'session data' is omitted; this is defined as data in terminal sessions and in forwarded channels (TCP, X11, and authentication agent). This will usually substantially reduce the size of the resulting log file. This option is Off by default.

NOTE: Not all SSH servers work properly. Various existing servers have bugs in them, which can make it impossible for a client like CETerm to talk to them unless it knows about the bug and works around it. Since most servers announce their software version number at the beginning of the SSH connection, CETerm will attempt to detect which bugs it can expect to see in the server and automatically enable workarounds.

The following configuration options are provided to navigate around these known bugs in the various SSH server implementations.

Bug – SSH-1 Ignore

Within the SSH-1 protocol, the client or server can send an “ignore message” at any time. Either side is required to ignore the message whenever it receives it. Within CETerm, this capability is used to hide the password packet in SSH-1, so that a listener cannot tell the length of the user's password. CETerm also uses “ignores messages” for application level keepalives. Certain SSH-1 servers lock up in using “ignore messages”.

If this option is not enabled, CETerm will assume that the SSH-1server does not have this bug.

If this option is enabled, CETerm session connection will succeed, but keepalives will not work and the session might be more vulnerable to eavesdroppers than it could be.

If the option is auto-sensed, CETerm will detect the bug and stop using “ignore messages”. The default option is Auto-Sense.

Bug – SSH-1 Password Hiding

When talking to an SSH-1 server which cannot deal with ignore messages CETerm will attempt to disguise the length of the user's password by sending additional padding *within* the password packet. This is technically a violation of the SSH-1 specification, and so CETerm will only do it when it cannot use standards-compliant ignore messages as camouflage. In this sense, for a server to refuse to accept a padded password packet is not really a bug, but it does make life inconvenient if the server can also not handle ignore messages.

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

This option only applies to SSH-1 servers. The default option is Auto-Sense.

Bug – SSH-1 RSA Auth

Some SSH-1 servers cannot deal with RSA authentication messages at all. If Pageant is running and contains any SSH-1 keys, CETerm will automatically try

RSA authentication before falling back to passwords, so these servers will crash when they see the RSA attempt.

If this bug is auto-sensed, CETerm will go straight to password authentication. If this option is enabled when talking to a correct server, the session will succeed, but of course RSA authentication will be impossible.

This option only applies to SSH-1 servers. The default option is Auto-Sense.

Bug – SSH-2 HMAC Key

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

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

This option only applies to SSH-2 servers. The default option is Auto-Sense.

Bug – SSH-2 Encryption Key

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

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

This option only applies to SSH-2 servers. The default option is Auto-Sense.

Bug – SSH-2 RSA Signature

Versions below 3.3 of OpenSSH require SSH-2 RSA signatures to be padded with zero bytes to the same length as the RSA key modulus. The SSH-2 draft specification says that an unpadded signature MUST be accepted, so this is a bug. A typical symptom of this problem is that CETerm mysteriously fails RSA authentication once in every few hundred attempts, and falls back to passwords.

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

This option only applies to SSH-2 servers. The default option is Auto-Sense.

Bug – SSH-2 ID in PK Auth

Versions below 2.3 of OpenSSH require SSH-2 public-key authentication to be done slightly differently: the data to be signed by the client contains the session ID formatted in a different way. If public-key authentication mysteriously does not work but the Event Log thinks it has successfully sent a signature, it might be worth enabling the workaround for this bug to see if it helps.

If this bug is auto-sensed, CETerm will sign data in the way OpenSSH expects. If this option is enabled when talking to a correct server, SSH-2 public-key authentication will fail.

This option only applies to SSH-2 servers. The default option is Auto-Sense.

Bug – SSH-2 Re-Key

Some SSH servers cannot cope with repeat key exchange at all, and will ignore attempts by the client to start one. Since CETerm pauses the session while performing a repeat key exchange, the effect of this would be to cause the session to hang after an hour (unless you have your rekey timeout set differently). Other, very old, SSH servers handle repeat key exchange even worse, and disconnect upon receiving a repeat key exchange request.

If this bug is auto-sensed, CETerm will never initiate a repeat key exchange. If this option is enabled when talking to a correct server, the session should still function, but may be less secure than you would expect.

This option only applies to SSH-2 servers. The default option is Auto-Sense.

Username

This is the field where you can specify what user name you want to login as, when connecting through your SSH server. Configuring a username will prevent you from having to explicitly type this on every connection.

The default is blank.

SSH Keep Alive

This is the time interval, in seconds, that CETerm will use for triggering SSH level keep-alive frames. Note SSH Keep-Alives are different from TCP protocol Keep Alives. A value of 0 implies not to use SSH Keep-Alives.

The default value is 0.

Log Mode

CETerm can maintain a log of all data interaction and exchange performed over the SSH connection. This can be helpful in troubleshooting connection failures. Use this setting to configure the type of log that CETerm should save. The amount of information saved in the log varies with the configured log mode setting. The log file is created in the root directory with the following name format: /ssh_log_&h.raw where &h is replaced by the hostname. The default mode is None.

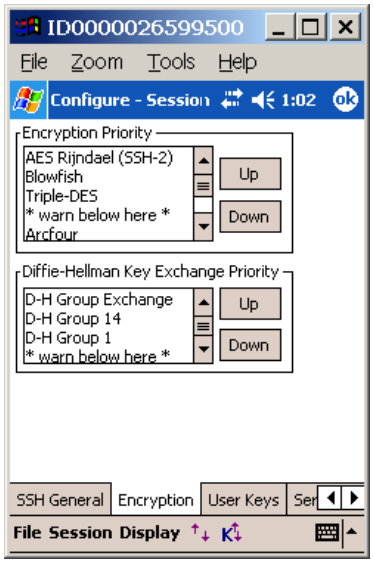
- None
- Info
- Debug
- SSH Data
- SSH Raw

Notices

This button displays SSH specific copyright notice.

ENCRYPTION

CETerm supports a variety of different encryption algorithms, and allows you to prioritize which one you prefer to use. Use this configuration tab to set a priority preference for the SSH encryption algorithms.



Encryption Priority

Highlight the preferred encryption algorithm and use the up and down buttons to position it in the list box to specify a priority preference order. When you make an SSH connection, CETerm will search down the list from the top until it finds an algorithm supported by the server, and then use that. By default, CETerm list the following encryption algorithms in priority order:

- AES Rijndael (SSH-2)
- Blowfish
- Triple-DES
- * warn below here *
- Arcfour
- * ignore following *
- DES

If the encryption algorithm which CETerm finds is below the ‘warn below here’ line, you will see a warning box when you make the connection:

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

This warns you that the first available encryption is not a very secure one. Typically you would put the “* warn below here *” line between the encryptions you consider secure and the ones you consider substandard. By default,

CETerm supplies a preference order intended to reflect a reasonable preference in terms of security and speed.

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

Any algorithms below the “* ignore following *” selection are not used and ignored by CETerm.

NOTE: Single-DES is not recommended in the SSH-2 draft protocol standards, but one or two server implementations do support it.

Diffie-Hellman Key Exchange Priority

Key exchange occurs at the start of an SSH connection (and occasionally thereafter, depending upon your settings in the SSH General tab); it establishes a shared secret that is used as the basis for all of SSH security features. It is therefore very important for the security of the connection that the key exchange is secure.

Key exchange is a cryptographically intensive process; if either the client or the server is a relatively slow machine, the slower methods may take several tens of seconds to complete.

NOTE: If connection startup is too slow, or the connection hangs periodically, you may want to try changing these settings. If you don't understand what any of this means, it's safe to leave these settings alone.

CETerm supports a variety of SSH-2 key exchange methods, and allows you to choose which one you prefer to use. This configuration is similar to encryption algorithm cipher selection. CETerm currently supports the following varieties of Diffie-Hellman key exchange:

- *D–H Group exchange*: with this method, instead of using a fixed group, CETerm requests that the server suggest a group to use for key exchange; the server can avoid groups known to be weak, and possibly invent new ones over time, without any changes required to CETerm's configuration. We recommend use of this method, if possible.

- *D–H Group 14*: a well-known 2048-bit group.

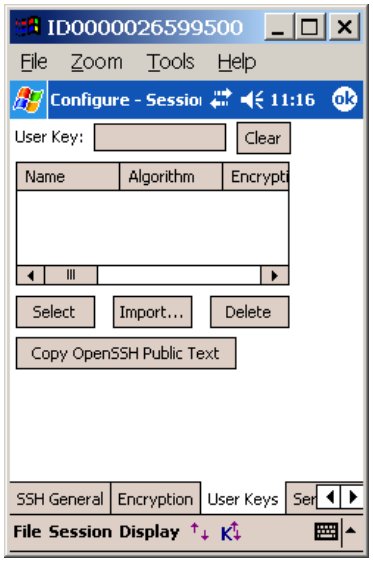
- *D–H Group 1*: a well-known 1024-bit group. This is less secure than group 14, but may be faster with slow client or server machines, and may be the only method supported by older server software.

If the first algorithm CETerm finds is below the "* warn below here*" line, you will see a warning box when you make the connection, similar to the one discussed in the previous (encryption priority selection) configuration.

USER KEYS

This configuration tab manages the User Keys to be used for authentication with the SSH server. User Keys are used for public key authentication. Public key authentication requires a key-pair consisting of a public key and a private key. The public key can be known be everybody whereas the private key is a closely held secret and is usually encrypted with a corresponding passphrase.

The public key is copied to the server and the private key is imported into CETerm. The private key is stored in CETerm in the encrypted form. The server and CETerm use the keys to authenticate the login request.



User Key

This is the key which has been selected for use with the current session. It can only be selected from the keys which have been imported into CETerm.

Clear

This will remove any currently selected key for the session. Without a key, the SSH connection will attempt to use password or other authentication mechanisms.

Select

This button selects the highlighted key in the table to be used as the User Key for this session.

Import

Tap this button to import a key into CETerm. CETerm can import keys generated for OpenSSH and ssh.com servers and some SSH client tools.

Delete

Tapping this button will delete the highlighted key from the table and remove it from CETerm settings.

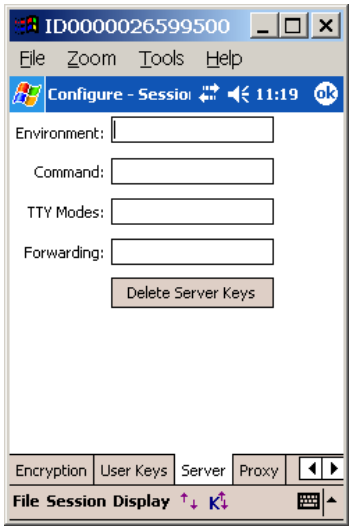
Copy OpenSSH Public Text

Tapping this button will copy the public text for the highlighted key into the device cut-and-paste clipboard and display the text in a popup window. This text is commonly put in the "authorized_keys" file in the user's home directory on the server.

The table contains a list of keys that have been imported into CETerm. These keys are shared by all CETerm sessions. Each key has a "Name" which is assigned by the user when the key is imported. Also shown is the "Algorithm" the key supports, the "Encryption" used for the key, and the "Comment" field of the key.

SERVER

The configuration attributes on this tab allow you to configure server options.



Environment

This setting specifies environment variables to be set on the SSH server. Not all servers will accept new environment variables. The format of the variables is a list of semicolon delimited name-value pairs:

```
name="value";name2=!a=b;c="d"!
```

Each value is delimited by quoting characters. Typically that character will be the double-quote (“). If the value contains double-quote characters, any other printable character may be used, including the single-quote or exclamation mark. Note that the value for name2 contains equal signs, double-quotes, and a semicolon and is delimited by the exclamation mark (!).
Whatever character is used at the start of the value must be used at the end.

The default setting is blank.

Command (Advanced)

This represents a special command or subsystem to invoke on the SSH server in lieu of an interactive shell. This is typically used for non-interactive host sessions. For most users this will be blank.

TTY Modes (Advanced)

This setting can be used to add TTY Modes to be sent to the SSH server. The format of the variables is a list of semicolon delimited name-value pairs:

```
mode="value";mode2="value2"
```

Each value is delimited by quoting characters. See Environment above for details on quoting.

Forwarding (Advanced)

This setting defines port forwarding or tunnels supported by this connection. Each tunnel is defined in the following format:

```
[4,6,A][L,R,D][sourcehost:]sourceport=desthost:destport;…
```

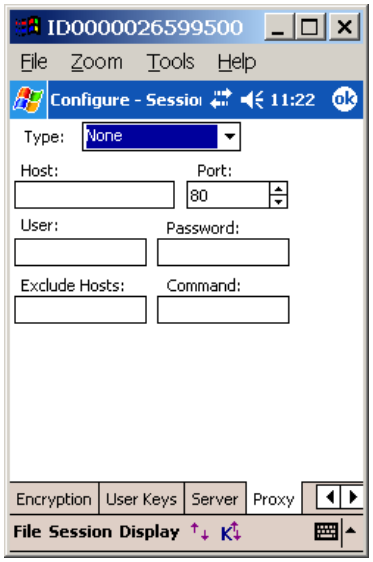
Where brackets indicate optional items,
4 – use IPV4, or 6 – use IPV6, or A – autodetect
L – local port forwarded, R – remote port forwarded, D – dynamic (proxy) port
For most users this will be blank.

Delete Server Keys

Tapping this button will erase all “known hosts” server keys stored within CETerm. The user must re-accept all keys during future host key negotiations.

PROXY

The Proxy tab allows you to configure CETerm to use various types of proxy servers in order to make network connections. The settings affect the primary network connection from your CETerm SSH session, but also any extra connections made as a result of SSH port forwarding.



Type

This option allows you to configure what type of proxy you want CETerm to use for its network connections. The choices are:

- *None*: No proxy is used.
- *SOCKS 4* or *SOCKS 5*: proxy through a SOCKS server.
- *HTTP*: proxy through a web server supporting the HTTP CONNECT command
- *Telnet*: Many firewalls implement a less formal type of proxy in which a user can make a Telnet connection directly to the firewall machine and enter a command such as connect myhost.com 22 to connect through to an external host. Selecting 'Telnet' allows you to tell CETerm to use this type of proxy.

The default setting is None.

Host

This is the DNS name or IP address of the proxy server. The default is blank.

Port

This is the port on which the proxy server is listening. Set this to match the port on the proxy server for connections. The default is 80.

User

If your proxy server requires authentication, enter the username. The default is blank.

Password

If your proxy server requires authentication, enter the password. The default is blank.

WARNING: This password is stored in plain text within CETerm.

NOTE: Authentication is not fully supported for all forms of proxy. Username and password authentication is supported for HTTP proxies and SOCKS 5 proxies.

With SOCKS 5, authentication is via CHAP if the proxy supports it otherwise the password is sent to the proxy in plain text.

With HTTP proxy, the only currently supported authentication method is 'basic', where the password is sent to the proxy in plain text.

SOCKS 4 can use the 'Username' field, but does not support passwords.

You can specify a way to include a username and password in the Telnet proxy command

Exclude Hosts

Typically you will only use a proxy to connect to non-local parts of your network. For example, your proxy might be required for connections outside your company's internal network. Use this setting to enter ranges of IP addresses, or ranges of DNS names, for which CETerm will avoid using the proxy and make a direct connection instead.

This setting may contain more than one exclusion range, separated by commas. Each range can be an IP address or a DNS name, with a ***** character allowing wildcards. For example:

`*.somehost.com`
excludes any host with a name ending in `.somehost.com` from proxying.

`192.168.88.*`

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

`192.168.88.*, *.somehost.com`
This excludes both of the above ranges at once.

Command

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

In this string, you can use `\n` to represent a new-line, `\r` to represent a carriage return, `\t` to represent a tab character, and `\x` followed by two hex digits to represent any other character. `\\` is used to encode the **** character itself.
Also, the special strings `%host` and `%port` will be replaced by the host name and port number you want to connect to. The strings `%user` and `%pass` will be replaced by the proxy username and password you specify. To get a literal `%` sign, enter `%%`.

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

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

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