TCP - Transmission Control Protocol (Client uses Telnet to log into Server)

Client Node		Internet	Server Node			EventHelix.com/EventStudio 1.0
Client		Net	Server			
User	Telnet	Net	Telnet Server	Telnet Thread	Shell	09-May-02 23:39 (Page 1)

Listen on Telnet Port (23)

Telnet server listens to TCP port 23

User initiates Telnet

User types telnet <IP Address>

User invokes the telnet application

create

Prints: Trying to connect to <IP address>

SYN

Telnet establishes TCP connection with the requested server

SYN_ACK

create

Telnet server is designed to handle multiple telnet sessions. Thus when a telnet connection request is received, it creates a thread to handle the individual telnet session. This way the telnet server is free to listen to port 23 for more telnet requests

Listen to Socket

The Telnet thread listens to the socket assigned for the telnet session

ACK

Prints: connected to <IP Address>

Telnet notifies the user that it has established the TCP connection with the destination machine.

Negotiation of Terminal Options

The communication between client and server is handled with internal commands, which are not accessible by users. All internal TELNET commands consist of 2 or 3-byte sequences, depending on the command type. The negotiation takes placing using such commands. Commands begin with the Interpret As Command (IAC) character. IAC is defined as 255. When IAC is received in a telnet stream, the receiver interprets the next one or two bytes as command.

Telnet uses "will", "won't", "do" and "don't" commands to negotiate options between the client and server. "Will" shows desire to use, or confirmation of using, the option indicated by the code immediately following. "Won't" shows refusal to use or continue to use the option. "Do" requests that other party uses, or confirms that you are expecting the other party to use, the option indicated by the code immediately following. "Don't" demands that the other party stop using, or confirms that you are no longer expecting the other party to use, the option indicated by the code immediately following.

do_SUPRESS_GO_AHEAD

Half duplex terminals operated with only transmission in one direction. Thus a go ahead signal was used to signal to the other end that it can transmit. Modern terminals are full duplex so they can work without a go ahead character. "Suppress go ahead" requests the other end to not use use the go ahead signal

will_TERMINAL_TYPE

reply

Telnet client informs the server that it wishes to identify terminal type

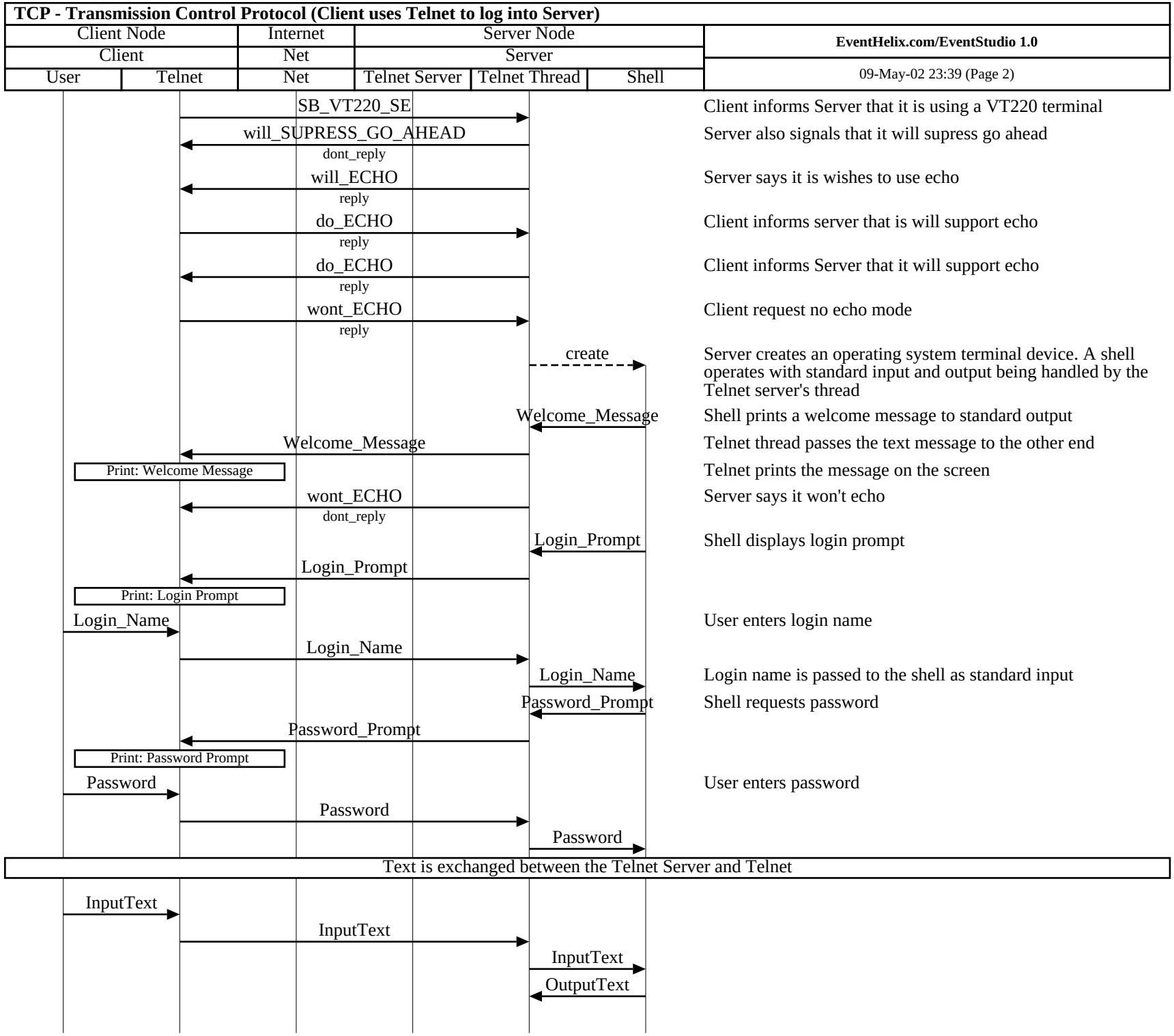
do_TERMINAL_TYPE

dont_reply

Server responds with the indication that it can handle terminal type identification

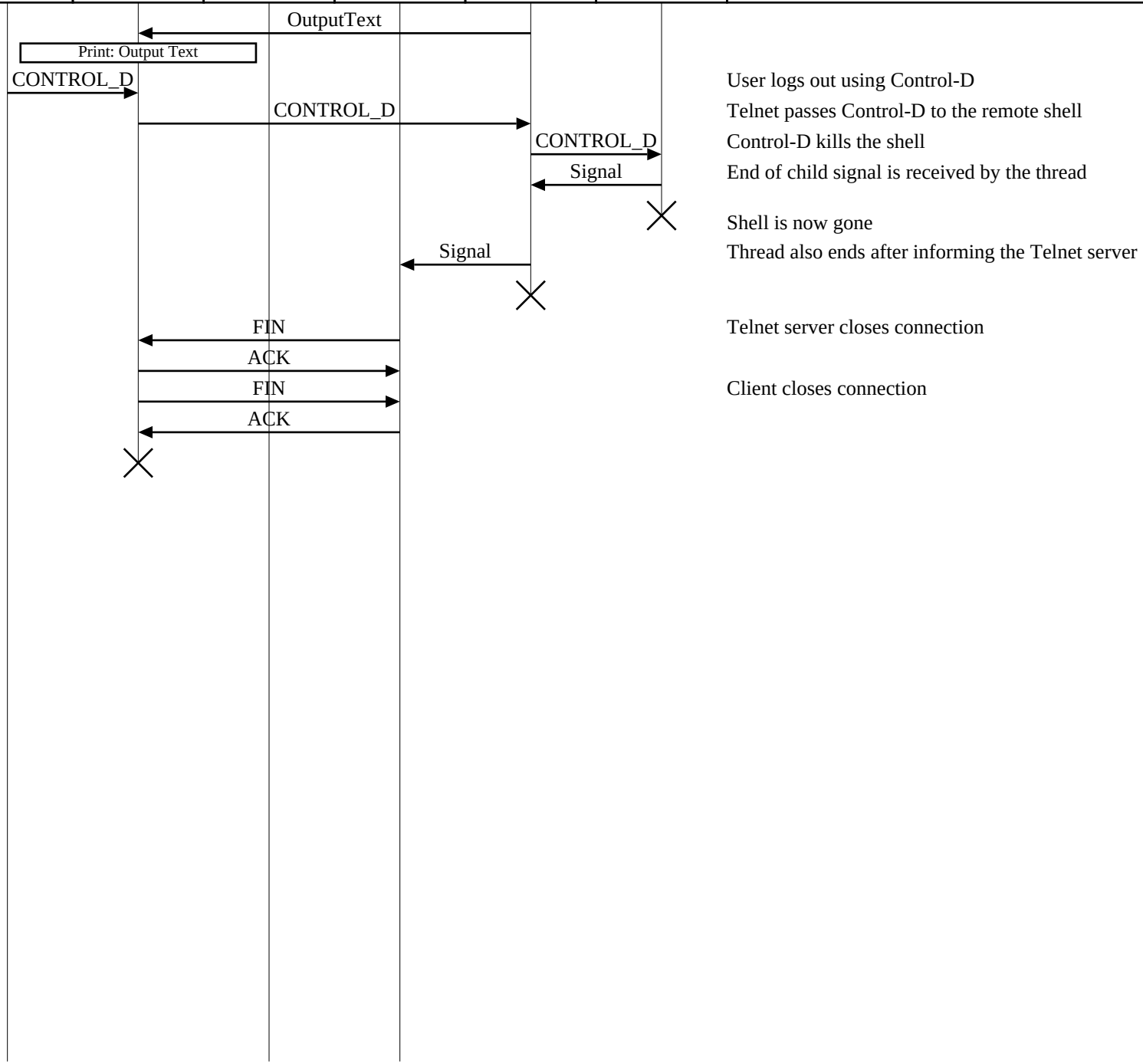
SB_REQUEST_TYPE_SE

Server requests terminal type. SB and SE commands enclose the sub-negotiation bytes


TCP - Transmission Control Protocol (Client uses Telnet to log into Server)
Client Node
Client
User
Telnet
Internet
Net
Server Node
Server
Net
Telnet Server
Telnet Thread
Shell
EventHelix.com/EventStudio 1.0
09-May-02 23:39 (Page 2)
SB_VT220_SE
Client informs Server that it is using a VT220 terminal
will_SUPRESS_GO_AHEAD
Server also signals that it will supress go ahead
dont_reply
will_ECHO
Server says it is wishes to use echo
reply
do_ECHO
Client informs server that is will support echo
reply
do_ECHO
Client informs Server that it will support echo
reply
wont_ECHO
Client request no echo mode
reply
create
Server creates an operating system terminal device. A shell operates with standard input and output being handled by the Telnet server's thread
Welcome_Message
Shell prints a welcome message to standard output
Welcome_Message
Telnet thread passes the text message to the other end
Print: Welcome Message
Telnet prints the message on the screen
wont_ECHO
Server says it won't echo
dont_reply
Login_Prompt
Shell displays login prompt
Login_Prompt
Print: Login Prompt
Login_Name
User enters login name
Login_Name
Login_Name
Login name is passed to the shell as standard input
Password_Prompt
Shell requests password
Password_Prompt
Print: Password Prompt
Password
User enters password
Password
Password
Text is exchanged between the Telnet Server and Telnet
InputText
InputText
InputText
OutputText

TCP - Transmission Control Protocol (Client uses Telnet to log into Server)

Client Node		Internet	Server Node			EventHelix.com/EventStudio 1.0
Client		Net	Server			
User	Telnet	Net	Telnet Server	Telnet Thread	Shell	09-May-02 23:39 (Page 3)



User logs out using Control-D

Telnet passes Control-D to the remote shell

Control-D kills the shell

End of child signal is received by the thread

Shell is now gone

Thread also ends after informing the Telnet server

Telnet server closes connection

Client closes connection