

iChip Telnet Client

Theory of Operation

Version 1.41

Pub. No.: 19-1500-00

International:
Connect One Ltd.
20 Atir Yeda Street, Kfar Saba 44643, Israel
Tel: +972-9-766-0456 Fax: +972-9-766-0461
info@connectone.com www.connectone.com

USA:
Connect One Semiconductors, Inc.
560 S. Winchester Blvd., Suite 500, San Jose, CA 95128
Tel: 408-572-5675 Fax: 408-572-5601
info@connectone.com www.connectone.com

Connect One
The Device Networking Authority

Introduction

General

The purpose of the TELNET Protocol is to provide a fairly general, bi-directional, eight-bit byte oriented communications facility. Its primary goal is to allow a standard method of interfacing terminal devices and terminal-oriented processes to each other. The protocol may also be used for terminal-terminal communication and process-process communication.

What is Telnet and what can it do for me?

Telnet is a protocol, which allows computers to communicate over a TCP/IP network (such as the Internet). When you Telnet in to a remote computer what you are actually doing is logging in and issuing commands as though you were sitting at the remote computer. Computer users on local area networks and on the Internet use Telnet to connect to a remote host and execute programs on the host.

Telnet server is a server application that runs on a host computer and allows remote users to Telnet in to a host. Telnet client is a client application that allows a user to Telnet out from his system to run a program on remote hosts.

Telnet is a useful way for system administrators, programmers and DOS legacy application users to Remotely administer a system, Run old Dos applications or Run programs where they run most efficiently (run programs on the host rather than on their machine, which It is more efficient to do in case the program requires large amounts of resources that reside on the host.).

Using telnet, a user can run character-based applications only. Telnet does not support graphical programs that use windows. It runs any program that runs in your Windows NT Command Shell Window.

Telnet specifies a variety of options that can be negotiated between a telnet client and server using commands at any stage during the connection. Options are agreed by a process of negotiation, which results in the client and server having a common view of various extra capabilities that affect the interchange and the operation of applications.

Telnet Login

A Telnet session starts by establishing connection to a remote system. Usually an authentication process will follow, that is usually made of user name and password prompts and replied, and then the telnet client will act as a dumb-terminal to the remote system, emulating a terminal type (typically vt100). It is then possible to issue commands.

The Telnet client component in iChip Family

iChip TELNET Client Feature Set

- Open TELNET link to TELNET Server
- Retrieve Data from Server
- Send A CRLF Terminated Line to the Server
- Send Binary Data to the Server
- Close TELNET Session

The telnet client component on iChip allows two operation modes:

1. iChip sends data to the remote Telnet server one line at a time. An AT+iTSND command is used to send a single (CR/LF terminated) line to the Telnet server.
2. iChip buffers data meant to be sent to the remote Telnet server and only sends it to the server when the host instructs it to send it or when the TCP/IP connection MTU is met. Data is sent as is, as a binary transmission.

Data is retrieved from the remote Telnet server as it is made available. Embedded Telnet options in the server's response stream are stripped by iChip before being turned over to the host.

The iChip will restrict its operation mode to the minimum implementation to assure best intersystem compatibility.

The following are the only Telnet options that will be negotiated by iChip:

Option ID	Name	Val	RFC
0	Binary Transmission	OFF	856
1	Echo	OFF	857
3	Suppress go ahead	Suppress	858
24	Terminal type	VT100	1091
31	Window size	Whatever	1073

1. Binary Transmission – iChip defaults to sending line mode CRLF terminated lines. iChip will ask the server to start binary transmission according to the host's commands (See also "Telnet binary send feature").
2. Echo – iChip will not echo incoming characters to the server.
3. Suppress go ahead - iChip will not send "go ahead" sequences to the server to indicate when the server can send the next character. iChip allows the server to send characters at any time.
4. Terminal type – The telnet server will assume iChip implements a VT100 terminal. iChip will transfer the VT100 control character to the host with the incoming data, which will allow the host to use them to implement a VT100 terminal emulation.
5. Window size – iChip replies that it *won't* negotiate this option.

Any other options negotiated by the Telnet server will be rejected by iChip.

Telnet binary Send Feature

(RFC 856 - TELNET BINARY TRANSMISSION)

When opening a telnet session, iChip assumes the default telnet settings for the binary transmission option, which is binary transmission off.

When binary transmission is off iChip will only allow TSND commands in order to send data to the telnet server and will always send CRLF terminated ASCII lines.

If the server asks iChip to activate the binary transmission option iChip will agree and change to binary mode.

In binary mode only TBSN and TFSH commands will be allowed for sending data to the telnet server and iChip will send the data as is (With the following exception - according to the RFC in this mode 0xFF characters must be doubled, in order to separate them from an 0xFF character that start negotiation of a telnet option or command).

When processing a telnet send command that is not allowed in the current state, iChip's telnet client will ask the telnet server to change the binary transmission mode. If the server agrees, iChip will change its mode and send the data, otherwise I/ERROR will be returned and the telnet client will not change its mode.

Telnet Error Management

Following is a list of the error codes that can be received in response to iChip TELNET commands, and some of the possible causes for each error:

550	<i>Telnet server not found</i>
551	<i>Timeout when connecting to Telnet server</i>
552	<i>Telnet command could not be completed</i>
553	<i>Telnet session shutdown by remote server</i>
554	<i>A Telnet session is not currently active</i>
555	<i>A Telnet session is already open</i>
556	<i>Telnet server does not allow operating in binary mode</i>
557	<i>Telnet server does not allow operating in ASCII mode</i>

TOPN

550 Telnet server not found: possible causes: The Telnet server name is a symbolic name but the iChip DNS settings are bad or the defined DNS servers are down and the Telnet server IP address cannot be resolved.

551 Timeout when connecting to Telnet server: iChip attempted to but was not able to open a socket connection to the Telnet server. Possible causes: The Telnet server name/IP address is incorrect, such a server does not exist, The iChip requires the use of a gateway to connect to the Telnet server but the gateway settings are incorrect or the gateway server is currently down.

555 A Telnet session is already open. Since only one Telnet session is allowed, the open Telnet session must be closed (TCLS) before opening a new Telnet session.

TSND

557 Telnet server does not allow operating in ASCII mode. Possible causes: Binary transmission was activated before the TSND command was processed (either because the Telnet server requested it or because of a previous TBSN command) and the telnet server did not allow turning it off. The command sends ASCII CRLF terminated lines, but in binary mode different interpretations of the CRLF might apply and so the data cannot be sent.

TBSN/TBSN%

556 Telnet server does not allow operating in binary mode. IChip asked but the Telnet server did not give iChip permission to transmit binary data and so the data cannot be sent. Possible cause: The Telnet server does not support transferring binary data.

TFSH/TFSH%

552 Telnet command could not be completed. Possible cause: The Telnet socket was closed by the server or connection between iChip and the Telnet server was lost before all the data could be sent or acknowledged.

TRCV/TSND/TBSN/TBSN%

553 Telnet session shut down by remote server. Possible causes: Telnet server inactivity timeout, internal error on FTP server.

TRCV/TSND/TBSN/TBSN%/TFSH/TFSH%/ TCLS

(All telnet commands but TOPN)

554 A Telnet session is not currently active. A Telnet session must be opened before this command can be executed.

Telnet Session Flow

Examples

1. Line at a time Telnet session (Telnet server opens command prompt):

- Step 1: Open Telnet session (TOPN)
- Step 2: Receive a login prompt from the server (TRCV)
- Step 3: Send the login name to the server (TSND)
- Step 4: Receive a password prompt from the server (TRCV)
- Step 5: Send the login password to the server (TSND)
- Step 6: Receive a command prompt from the server (TRCV)
- Step 7: Send a command line to the server (TSND)
- Step 8: Receive a reply from the server (TRCV)
- Step 9: Repeat steps 7 and 8 until no more lines to send/receive
- Step 10: When done, close the Telnet session (TCLS)

2. Binary Telnet session (Telnet server opens an application):

- Step 1: Open Telnet session (TOPN)
- Step 2: Receive a login prompt from the server (TRCV)
- Step 3: Send the login name to the server (TSND)
- Step 4: Receive a password prompt from the server (TRCV)
- Step 5: Send the login password to the server (TSND)
- Step 6: Receive a command/request from the application run by the server (TRCV)
- Step 7: Send a reply or data to the server (TBSN)
- Step 8: Repeat step 7 until no more data to send
- Step 9: Flush the data that was not sent to the server yet (TFSH)
- Step 10: Repeat steps 6, 7, 8 and 9 until no more data to send or receive
- Step 11: When done, close the Telnet session (TCLS)

Example of a Telnet Session

Below are examples of Telnet sessions.

Example 1: In this example the device logs in to a remote Telnet server. The device logs in using User name: “myuser” and Password: “mypass”. The device then operates a process on the server called “managedevice.exe”. The process sends two types of commands to the device:

1. “GET <n>” – The process prompts the device for the number of bytes of information it wishes to collect from the device at that point (n bytes).
2. “SET <parameter name> <parameter value>” – The process sends an instruction to the device to set one of its parameters to a new value (to effect its future operation). In this case the device echoes the set request as a confirmation that the action was performed.

Example 2: In this example the device logs in to a remote Telnet server. The device logs in using User name: “myuser” and Password: “mypass”. The TELNET server automatically starts a data collecting application on login. After a short welcome message the application expects to receive data records of 10 bytes at a time. A sequence of 10 *’s indicates the last data record has been sent. The device sends the data to iChip as it is made available in the device (one byte at a time), to avoid using its own memory to save it until 10 bytes are available. After 10 bytes are available the device instructs iChip to send them to the application on the TELNET server.

Example 1:

AT+itopn: telnetserver.com I/OK	<i>Step 1: Open a Telnet session to the Telnet server, using the Telnet server name</i>
AT+itrev I/6 login: AT+itsnd:myuser I/OK AT+itrev I/9: password: AT+itsnd:mypass I/OK AT+itrev	<i>Step 2: Authenticate wait for the login prompt wait for the password prompt send password wait for authentication process to finish and the first command prompt to appear</i>
I/3: C:\	<i>Step 3: Start the management application on the remote server</i>
AT+itsnd:"managedevice.exe" I/OK	<i>Step 4: Read data stream and perform the action requested</i>
AT+itrev I/8 GET 10 AT+itbsn%:10:0123456789	<i>a data prompt for 10 bytes - provide the bytes(*)</i>
I/OK	<i>Binary mode was activated by iChip successfully and the data was sent to the server</i>
AT+itrev I/9 SET A 7 AT+itsnd:A 7 <i>request</i>	<i>perform the "SET" and acknowledge the</i>
I/OK	<i>Binary mode was disabled by iChip successfully and the data was sent to the server followed by CRLF</i>
AT+itcls I/ONLINE	<i>Step 5: Close the Telnet session</i>

Example 2:

AT+itopn: telnetserver.com
I/OK

AT+itrcv
I/6
login:
AT+itsnd:myuser
I/OK
AT+itrcv
I/9:
password:
AT+itsnd:mypass
I/OK
AT+itrcv

I/85:
Welcome to your data collector
Please send data
Send *** to indicate end of data**

AT+itbsn: 1:0
I/OK

Step 1: Open a Telnet session to the Telnet server, using the Telnet server name

Step 2: Authenticate
wait for the login prompt

wait for the password prompt

send password

wait for authentication process to finish and the welcome message to appear

let iChip buffer the currently available data
Binary mode was activated by iChip successfully
and the data are buffered in iChip's memory

AT+itbsn:1:1 I/OK	<i>let iChip buffer the currently available data</i>
AT+itbsn:1:2 I/OK	<i>let iChip buffer the currently available data</i>
AT+itbsn:1:3 I/OK	<i>let iChip buffer the currently available data</i>
AT+itbsn:1:4 I/OK	<i>let iChip buffer the currently available data</i>
AT+itbsn:1:5 I/OK	<i>let iChip buffer the currently available data</i>
AT+itbsn:1:6 I/OK	<i>let iChip buffer the currently available data</i>
AT+itbsn:1:7 I/OK	<i>let iChip buffer the currently available data</i>
AT+itbsn:1:8 I/OK	<i>let iChip buffer the currently available data</i>
AT+itbsn:1:9 I/OK	<i>let iChip buffer the currently available data</i>
At+itfsh	<i>10 bytes are available - instruct iChip to send the buffered data</i>
I/OK	<i>data was sent to the TELNET server. Optionally the character % could be added to the TFSH command to promise the I/OK reply will only be returned after the server acknowledged receiving the data (in TCP level).</i>
At+itbsn%:10:***** I/OK	<i>send the indication that all data has been sent</i>
AT+itcls	Step 5: <i>Close the Telnet session</i>
I/ONLINE	