

Telnet Specifications

@version@ (@date@)

by Dieter Wimberger

Table of contents

1 About.....	2
2 Telnet Protocol Specifications.....	2
3 Implementation Comments.....	2
3.1 General.....	2
3.2 Specific.....	3

1. About

This document provides links to telnet specifications, and describes the implementation status of this library.

2. Telnet Protocol Specifications

[RFC 854](http://www.faqs.org/rfcs/rfc854.html) (<http://www.faqs.org/rfcs/rfc854.html>)

Telnet Protocol Specification

[RFC 855](http://www.faqs.org/rfcs/rfc855.html) (<http://www.faqs.org/rfcs/rfc855.html>)

Telnet Option Specifications

[RFC 857](http://www.faqs.org/rfcs/rfc857.html) (<http://www.faqs.org/rfcs/rfc857.html>)

Telnet Echo Option

[RFC 858](http://www.faqs.org/rfcs/rfc858.html) (<http://www.faqs.org/rfcs/rfc858.html>)

Telnet Suppress Go Ahead Option

[RFC 727](http://www.faqs.org/rfcs/rfc727.html) (<http://www.faqs.org/rfcs/rfc727.html>)

Telnet Logout Option

[RFC 1073](http://www.faqs.org/rfcs/rfc1073.html) (<http://www.faqs.org/rfcs/rfc1073.html>)

Telnet Window Size Option

[RFC 1091](http://www.faqs.org/rfcs/rfc1091.html) (<http://www.faqs.org/rfcs/rfc1091.html>)

Telnet Terminal-Type Option

[RFC 1184](http://www.faqs.org/rfcs/rfc1184.html) (<http://www.faqs.org/rfcs/rfc1184.html>)

Telnet Linemode Option

[RFC 1143](http://www.faqs.org/rfcs/rfc1143.html) (<http://www.faqs.org/rfcs/rfc1143.html>)

Telnet Environment Option

[RFC 1572](http://www.faqs.org/rfcs/rfc1572.html) (<http://www.faqs.org/rfcs/rfc1572.html>)

The Q Method of Implementing Option Negotiation

3. Implementation Comments

3.1. General

telnetd has been implemented based on the standard specifications, as well as the [Q Method of Implementing Option Negotiation \(RFC 1143\)](http://www.faqs.org/rfcs/rfc1143.html) (<http://www.faqs.org/rfcs/rfc1143.html>) .

Although it does some and could do more sophisticated option/suboption negotiation, the idea is to keep it down to the minimum required to work properly for the application you want to expose via telnet. At the moment the largest part of the negotiation is mainly limited to the beginning of the connection, this is specifically the case for ECHO and for LINEMODE, which means you will have to extend the code in case you require dynamic

Telnet Specifications

switching at throughout the connection.

Warning:

Although there exists a standard, telnet is a rather "complex" protocol which is implemented to different extent and with different success in various clients. Often, clients are "broken" or lack support for more complex negotiation/subnegotiation schemes.

Not always it is the telnetd implementation that has to be blamed. A good way to figure out what is going wrong is to enable debug in a standard UNIX telnet implementation (e.g. BSD telnet).

The following snippet shows how this can be achieved and what the output looks like:

```
[Fangorn:~] wimpi$ telnet
telnet> set options
Will show option processing.
telnet> open localhost 6666
Trying ::1...
Connected to localhost.
Escape character is '^]'.
RCVD WILL ECHO
SENT DO ECHO
RCVD DONT ECHO
RCVD DO NAWS
SENT WILL NAWS
SENT IAC SB NAWS 0 130 (130) 0 24 (24)
RCVD WILL SUPPRESS GO AHEAD
SENT DO SUPPRESS GO AHEAD
RCVD DO SUPPRESS GO AHEAD
SENT WILL SUPPRESS GO AHEAD
RCVD DO TERMINAL TYPE
SENT WILL TERMINAL TYPE
RCVD IAC SB TERMINAL-TYPE SEND
SENT IAC SB TERMINAL-TYPE IS "VT100"
```

3.2. Specific

NVT

The implementation has support for AYT and BRK. The first will be answered by writing directly to the terminal (which might cause troubles if you are using screens), the latter will trigger a connection event that can be handled by your application.

Options always negotiated

The implementation will always try to negotiate "SUPRESS GO AHEAD", "WINDOW SIZE", as well as "TERMINAL TYPE".

Character Mode

The implementation will negotiate server side only echo at the beginning of the connection. For the specification, IAC DONT ECHO would not be necessary, but

for some clients that have a broken ECHO option implementation.

Line Mode

The implementation will negotiate "LINEMODE" at the beginning of the connection. If the client does support and acknowledge linemode, subnegotiations about the mode mask and the special characters will take place. telnetd will subnegotiate the standard mode mask EDIT|TRAPSIG, as well as the special character set (sending or acknowledging defaults (as there is no actual support). If you need anything more sophisticated you will have to modify the existing implementation to add support for SLC as well as for the more complex subnegotiation to be expected.

Warning:

There is quite a number of clients which do not support line mode.