

Introduction to the Telnet protocol

The **Telnet** protocol is a standard internet protocol enabling terminals and applications to interface over the Internet. This protocol provides the basic rules making it possible to link a client (system composed of a display and keyboard) to a command interpreter (server side).

The Telnet protocol is applied on a [TCP](#) connection to send data in [ASCII](#) format coded over 8 bits between which the Telnet check sequences come. It therefore provides a communication orientated bi-directional system (half-duplex), coded over 8 bits and easy to implement.

The Telnet protocol relies on three basic concepts:

- The *Network Virtual Terminal* (NVT) paradigm;
- The negotiated options principle;
- The rules of negotiation.

This is a base protocol, to which certain other protocols from the TCP/IP suite (FTP, SMTP, POP3, ...) are applied. Telnet specifications do not mention authentication because Telnet is totally separated from applications which use it (FTP protocol defines an authentication sequence above Telnet). Additionally, the Telnet protocol is a non secure data transfer protocol, that is the data which it conveys circulates on the network in plain text (in an [unencrypted](#) way). When the Telnet protocol is used to connect a remote host to the machine upon which it is implemented as server, this protocol is assigned to [port 23](#).

Except for the associated options and negotiation rules, the Telnet protocol specifications are basic. Data transmission through Telnet consists only of transmitting bytes in the TCP flow (the Telnet protocol specifies that data must by default, i.e. if no option specifies to the contrary, be grouped in a buffer before being sent. More precisely this means that by default the data is sent line by line). When byte 255 is transmitted, the following byte must be interpreted as a command. Byte 255 is therefore called *IAC* (*Interpret As Command*). The commands are described further on in the document.

The basic specifications of the Telnet protocol are available in [RFC 854](#), while the many options are described in RFCs 855 to 861.

RFCs relating to Telnet	
RFC 854	Telnet Protocol Specifications
RFC 855	Telnet Option Specifications
RFC 856	Telnet binary transmission
RFC 857	Telnet Echo Option
RFC 858	Telnet Suppress Go ahead Option
RFC 859	Telnet Status Option
RFC 860	Telnet Timing Mark Option
RFC 861	Telnet Extended options-list Option

The notion of the virtual terminal

When the Internet began, the network (ARPANET) was made up of machines where configurations were slightly different (keyboards, character sets, resolutions, length of lines displayed). Moreover, the terminal sessions also had their own way of controlling the incoming/outgoing data flow.

So, instead of creating adaptors for each terminal type so that there could be interoperability between these systems, it was decided to develop a standard interface, called *NVT (Network Virtual Terminal)*, providing a standard communication base, comprised of:

- 7 bit ASCII characters to which the extended ASCII code is added
- Three control characters
- Five optional control characters
- A set of basic control signals

The Telnet protocol therefore consists of creating an abstraction of the terminal enabling whichever host (client or server) to communicate with another host without knowing its characteristics.

The negotiated options principle

The Telnet protocol specifications make it possible to take into account the fact that certain terminals can offer additional services, not defined in the basic specifications (but in accordance with the specifications), so as to be able to use advanced functions. This functionality is reflected in terms of options. The Telnet protocol therefore offers a system of option negotiations enabling the use of advanced functions in the form of options on either side by initiating requests for its authorisation from the remote system.

The Telnet options separately affect each direction of the data channel. So, each end is able to negotiate the options, i.e. to define the options that it:

- wants to use (*DO*)
- refuses to use (*DON'T*)
- wants the other end to use (*WILL*)
- refuses that the other end use (*WON'T*)

In this way, each party can issue a request for an option to be used. The other party must then respond as to whether or not it accepts the use of the option. Where the request relates to the deactivation of an option, the recipient of the request must not refuse so as to be completely compatible with the NVT model.

Telnet negotiated options		
Request	Response	Interpretation
DO	WILL	The originator begins by using the option
	WON'T	The originator must not use the option
WILL	DO	The originator begins by using the option, after having sent a <i>DO</i>
	DON'T	The originator must not use the option
DON'T	WON'T	The originator indicates that it has deactivated the option
WON'T	DON'T	The originator indicates that the originator must deactivate the option

There are 255 option codes. All the same, the Telnet protocol provides an address space allowing new options to be described.

RFC 855 explains how to document any new option.

The rules of negotiation

The rules of negotiation for options make it possible to avoid wraparound situations (for example when one of the parties sends option negotiation requests to each confirmation from the other party).

1. Requests must only be issued at the time of a mode change.
2. When one of the parties receives a mode change request, it must only acknowledge it if it is not already in the appropriate mode.
3. A request must only be inserted in the data flow at the place where it takes effect.

Exit control characters

The following characters are commands making it possible to control the network virtual terminal display:

Display control commands			
Number	Code	Name	Meaning
0	NULL	Null	This command allows data to be sent to the remote host without it being interpreted (in particular to indicate that the local host is still online)
1	LF	Line Feed	This command makes it possible to place the cursor on the following line, in the same horizontal position
2	CR	Carriage Return	This command makes it possible to place the cursor to the extreme left of the current line

So, the command *CFLF* is defined, composed of two commands *CR* and *LF* one after the other (in any order) making it possible to place the cursor at the extreme left on the following line.

Optional control characters

The previous characters are the only ones (among the 128 basic ASCII code characters and 128 extended ASCII code characters) to have a particular meaning for the network virtual terminal. The following characters can potentially have a meaning on a network virtual terminal but are not necessarily used.

Display control commands			
Number	Code	Name	Meaning
7	BEL	Bell	This command allows a sound or visual signal to be issued without changing the cursor position
8	BS	BackSpace	This command allows the cursor position to be changed to its previous position
9	HT	Horizontal Tab	This command allows the cursor position to be changed to the next tabbed position to the right
11	VT	Vertical Tab	This command allows the cursor position to be changed to the next tabbed position on the line below
12	FF	Form Feed	This command allows the cursor position to be changed to the bottom of the following page while retaining its horizontal position

Session control characters

The following characters are commands which make it possible to control the Telnet session. In order to be interpreted as such these commands must be preceded by the escape character *IAC* (*Interpret As Command*). If these bytes are transmitted without being preceded by the *IAC* character, they will be processed as simple characters. To transmit the *IAC* character, it must (itself) be preceded by an escape character, in other words it must be doubled.

The commands relating to an option negotiation must be followed by a byte specifying the option. These commands make it possible to interrupt signals, delete information in the terminal cache, etc.

session control characters			
Number	Code	Name	Meaning
240	SE		End of option negotiation
241	NOP	No Operation	This command allows data to be sent to the remote host without it being interpreted (in particular to indicate that the local host is still online)
242	DM	Data Mark	Makes it possible to empty all buffers between the virtual network terminal and the remote host. It relates to a push on the NVT Synch button and must be linked to a flagging of the Urgent TCP bit
243	BRK	Break	Break character for the virtual terminal
244	IP	Interrupt Process	This command makes it possible to suspend, interrupt or abandon the remote process
245	AO	Abort Output	This command makes it possible to suspend, interrupt or abandon the display of the remote process
246	AYT	Are You There	This command makes it possible to check that the remote system is still "alive"
247	EC	Erase Character	This command allows the previous character to be deleted.
248	EL	Erase Line	This command allows the previous line to be deleted.
249	GA	Go Ahead	This command makes it possible to reverse control, for half-duplex links
250	SB	SB	This command indicates that the data which follows is a negotiation for the previous option
251	WILL	option code	
252	WON'T	option code	
253	DO	option	

		code	
254	DON'T	option code	
255	IAC	Interpret As Command	This command allows the following byte to be interpreted as a command. The IAC command makes it possible to go beyond the basic commands