

Telnet Test Port for TTCN-3 Toolset with TITAN, Function Specification

Gábor Szalai

Version 155 17-CNL 113 320, Rev. J, 2012-05-15

Table of Contents

How to Read this Document	1
Scope	1
General	1
Interface Description	1
Client Mode	1
Server Mode	2
Function Specification	3
Configuration and Management	3
Capacity and Limitations	4
Platform Limitations	4
Server Side Limitations in Client Mode Operation	4
Terminology	5
Abbreviations	5
References	5

How to Read this Document

This is the Function Specification for the Telnet test port. The Telnet test port is developed for the TTCN-3 Toolset with TITAN.

Scope

The purpose of this document is to specify the functionality of the Telnet test port.

General

The purpose of the Telnet Test Port is to adapt TITAN abstract test components to the real test system interface. The Telnet Test Port makes it possible to use remote telnet login from TTCN-3 via the TCP layer of the operating system. The test port supports client and server mode operation either. In server mode operation the Telnet Test Port can handle one connection simultaneously.

The Telnet Test Port supports the capabilities of Network Virtual Terminal as described in [3].

Client mode operation: The supported commands are **IAC**, **WILL**, **WONT**, **DO**, **DONT**, **SB** and **SE**. Telnet options described in [4], [5] and [6] are supported. Unsupported options with **WILL** tag will be echoed to the host with **DONT** and options with **DO** tag will be echoed with **WONT** tag.

Server mode operation: The supported commands are **IAC**, **WILL**, **WONT**, **DO** and **DONT**. Telnet options are not supported. Options with **WILL** tag will be echoed to the host with **DONT** and options with **DO** tag will be echoed with **WONT** tag.

Interface Description

Client Mode

In client mode operation the Telnet Test Port uses charstring messages as an interface towards the TITAN test executor environment, and ASPs to control the test port behavior dynamically. Supported ASPs are listed in the [table](#) below.

See Supported ASPs in the table below:

Type	Description
<code>ASP_TelnetPortParameters</code>	The test port stores the parameters received in this ASP and applies them if the test port is remapped. Fields of this ASP are the same as the test port parameters. Omitted fields mean that the respective parameters will not change. This allows changing the test port parameters.

Type	Description
ASP_TelnetDynamicConfig	Allows changing the following test port parameters dynamically: read mode, window size, echo, prompt. Its effect is instantaneous.
ASP_TelnetConnection	Controls the connection. Currently supports one parameter: CONNECT . Can be used to re-establish the telnet connection (possibly with new test port parameters).

An "in" integer parameter is also used to signal any kind of notification by the test port to the user. Currently the following values are implemented listed in the [table](#) below:.

See Test port client notifications below:

Value	Description
0	Used for signalling that the connection is terminated by the server or that the attempt to create a connection failed. The use of this integer value is configurable by setting the CTRL_DETECT_SERVER_DISCONNECTED or parameter CTRL_DETECT_CONNECTION_ESTABLISHMENT_RESULT parameter to "yes" respectively. For further information see [2] .
2	Used for signalling that the attempt to create a connection succeeded. The use of this integer value is configurable by setting the parameter CTRL_DETECT_CONNECTION_

The Telnet Test Port is connected to the SUT/IUT via TCP connection.

Server Mode

In server mode operation the Telnet Test Port uses charstring messages as an interface towards the TITAN test executor environment, and an ASP to close the connection to the client.

An "in" integer parameter is also used to signal events to the user. Currently the following values are implemented listed in the [table](#) below.

See Test port server notification in the table below:

Value	Description
1	a successful login from the client with the username <code>CTRL_USERNAME_CLIENT</code> . This value is only sent to the user after the login procedure (i.e. the user must provide the correct password set in <code>CTRL_PASSWORD_CLIENT</code>) or, if the <code>CTRL_LOGIN_SKIPPED</code> parameter is set to the value "yes", "Yes" or "YES" in the runtime configuration file, immediately in case of incoming connection. For further information see [2].
3	Used for signalling that the connection is terminated by the client. The use of this integer value is configurable by setting the <code>CTRL_DETECT_CLIENT_DISCONNECTED</code> parameter to "yes". For further information see [2].

The Telnet Test Port is connected to the SUT/IUT via TCP connection. The supported ASP is:

Type	Description
<code>ASP_TelnetClose</code>	Closes the connection to the client. After sending this ASP the Test Port can accept further connections.

Function Specification

Configuration and Management

The runtime behavior of the Telnet Test Port can be configured via an external runtime configuration file.

The configuration file contains the following information for the Telnet Test Port:

In client mode operation:

- Host name of the SUT/IUT.
- TCP port number of the SUT/IUT.
- Username used for login.
- Password used for login.
- Domain used for login.
- Read mode (waiting for a newline in the received message or present every message immediately to TTCN when it is received)
- Possible prompt strings with or without wildcards
- Telnet terminal type

- Telnet echo option
- Telnet window size: height and width
- Whether to use CR LF or LF as linefeed (newline)
- Logging in user with or without user authentication
- Sending/not sending an integer value to the user when the connection is terminated by the server.
- Sending/not sending an integer value to the user about the result of the connection attempted.
- Enabling the filtering of linefeeds directly preceding the prompt from incoming messages.

In server mode operation:

- TCP port number for listening for connections
- Username used for logging the client in
- The 'login' prompt (the charstring to send to prompt the client for the login name)
- Password used for logging the client in
- The 'password' prompt (the charstring to send to prompt the client for the password)
- Prompt string
- Logging in user with or without user authentication
- Use the port in client or server mode operation
- Sending/not sending an integer value to the user when the connection is terminated by the client.
- Enabling the setting whether a failed message sending should result in an error or a warning.
- Enabling the attaching of the server prompt to the end of every outgoing message.

Capacity and Limitations

Platform Limitations

Client Mode

Only Solaris, Suse linux, RHEL 4 (Nahant Update 6) platforms and Pragma Telnet Server running on Windows Server 2003 have been tested.

Server Mode

Only Solaris and RHEL 4 (Nahant Update 6) platforms have been tested.

Server Side Limitations in Client Mode Operation

All the tested servers echoed characters regardless of the echo option. Because of this, the test port

contains a filtering mechanism for echoed characters. This mechanism is enabled if echo is disabled in the configuration file and vice versa. The telnet echo option (if enabled) is sent to the server regardless of the filter settings, but it depends on the value set in `CTRL_ECHO`.

The way telnet echo option is negotiated can cause problems with certain applications or servers. Because of this and because of the telnet echo option did not work with the servers tested, the test port is not compiled with the echo option by default, regardless of the setting of `CTRL_ECHO`. It can be enabled via the C++ compiler flag `TELNET_ECHO_OPTION` if needed. The echo filtering mechanism is still functional if this compiler flag is not defined.

The telnet test port can handle only one connection at the same time. Prior to a new connection the previous one has to be cleared. The size of the listening queue is 1. Thus one incoming connection might be in the pending state, meaning that a second connection will succeed, but no traffic is possible via this connection until the previous one is closed.

The term "echo option" is used for the Telnet echo option used during the negotiation of the set up of a Telnet session as defined in [4]. Not to be confused with the `CTRL_ECHO` "option", which is only described in [2].

Terminology

No specific terminology is used.

Abbreviations

ASP

Abstract Service Primitive

TTCN-3

Testing and Test Control Notation version 3

IUT

Implementation Under Test

SUT

System Under Test

TCP

Transmission Control Protocol

References

[1] ETSI ES 201 873-1 v.3.2.1 (2007-02)

The Testing and Test Control Notation version 3. Part 1: Core Language

[2] Telnet Test Port for TTCN-3 Toolset with TITAN, User Guide

[3] [RFC 854](#)

Telnet protocol specification

[4] [RFC 857](#)

Telnet echo option

[5] [RFC 1073](#)

Telnet Window Size Option

[6] [RFC 1091](#)

Telnet Terminal-Type Option