Prepared (also subject responsible if other)
ETH/RZX Endre Kulcsar +36 1 437 7469

No.
198 17-CNL 113 642 Uen

Approved	Checked	Date	Rev	Reference
ETH/RZXC (Elemér Lelik)		2009-04-20	A	GASK2

STDINOUT Test Port for TTCN-3 Toolset with TITAN, User Guide

Contents

1 Introduction

1.1 Revision history

Date	Rev	Characteristics	Prepared
2009-04-20	PA1	Initial version	ETHEKR

1.2 About this Document

1.2.1 How to Read this Document

This is the User Guide for the STDINOUT Test Port. The STDINOUT Test Port is developed for the TTCN-3 Toolset with TITAN [5]. This document is intended to be read together with Product Revision Information [2] and Functional Specification [3].

1.2.2 Presumed Knowledge

The knowledge of the TITAN TTCN-3 Test Executor [5] and the TTCN-3 language [1] is essential.

1.2.3 References

[1] ETSI ES 201 873-1 v3.2.1 (02/2007)
The Testing and Test Control Notation version 3. Part 1: Core Language

[2] 109 21-CNL 113 642-1
STDINOUT Test Port for TTCN-3 Toolset with TITAN, Product Revision Information

[3] 155 17-CNL 113 642
STDINOUT Test Port for TTCN-3 Toolset with TITAN, Function Specification

[4] 1/1531-CRL 113 200 Uen
Installation Guide for TITAN TTCN-3 Test Executor

[5] 1/19817-CRL 113 200 Uen
User Guide for TITAN TTCN-3 Test Executor

1.2.4 Abbreviations

SUT System Under Test

TP Test Port: Adaptation between TITAN TTCN-3 Test Executor and SUT.

TTCN-3 Testing and Test Control Notation version 3

1.2.5 Terminology

None.

1.3 System Requirements

In order to operate the STDINOUT Test Port the following system requirements must be satisfied:

- TITAN TTCN-3 Test Executor R7A (1.7.pl0) or higher installed. For installation guide see [4]. Please note: This version of the Test Port is not compatible with TITAN releases earlier than R7A.
- Unix, Sun Solaris or Linux operating system.

1.4 Fundamental Concepts

This Test Port handles connection between the TTCN-3 test executor and the operator.

2 The Test Port

2.1 Overview

The STDINOUT Test Port provides a simple interface between the TTCN-3 test suite and operator.

The operator can enter text in a terminal (stdin) and the test port transmits this text as a charstring to the TTCN-3 test suite.

The TTCN-3 test suite can send a charstring and the test port outputs this to the terminal (stdout).

2.2 Installation

Since the STDINOUT Test Port is used as a part of the TTCN-3 test environment this requires TTCN-3 Test Executor to be installed before any operation of the STDINOUT Test Port. For more details on the installation of TTCN-3 Test Executor see the relevant section of [4].

2.2.1 Preparation

The STDINOUT Test Port package contains the following files:

```
STDINOUTmsg_PT.cc
STDINOUTmsg_PT.hh
STDINOUTmsg_PortType.ttcn
```

2.2.2 Description of the files in the package

2.2.2.1 STDINOUTmsg_PortType.ttcn

This contains the STDINOUT Test Port definition.

2.2.2.2 STDINOUTmsg_PT.hh

This is the C++ header file of the STDINOUT Test Port

2.2.2.3 STDINOUTmsg_PT.cc

This is the C++ source file of the STDINOUT Test Port

2.3 Configuration

The executable test program behaviour is determined via the run-time configuration file. This is a simple text file, which contains various sections (e.g. [TESTPORT_PARAMETERS]) after each other. The usual suffix of configuration files is .cfg. For further information about the configuration file see [5].

2.3.1 Test Port parameters in the Test Port configuration file

No test port parameters are used for this Test Port.

2.4 Test Port Operation

The STDINOUT Test Port has no ASPs. The Test Port can be used for sending and receiving TTCN-3 charstrings.
The TTCN-3 “send” command followed by the necessary charstring in the TTCN-3 code will cause the text to be displayed at stdout (i.e.the terminal). The operator entered text at stdin (i.e. the terminal) is converted to TTCN-3 charstring by the test port when pressing “Enter”. The “receive” command in the TTCN-3 code has to be used for the TTCN-3 test suite to receive this charstring.

See the Examples section for example ways to use the STDINOUT Test Port in command line mode.

3 Error messages

"Only one STDINOUT Test Port entity can be mapped on the same time"

The TTCN-3 map function can be used only once on a component for this test Port. If the map function is repeated then this error message is displayed.

4 Examples

The demo directory includes the example TTCN-3 file STDINOUT_Test.ttcn. This file includes a simple test which outputs “Hello, world!” and expects the user input of “Hello, TTCN-3!”. An example Makefile, stdinout.cfg file and ttcn3_autostart.sh file is also included.

4.1 Non-parallel Execution Mode

When the Makefile includes “TTCN3_LIB = ttcn3”
then the user can run the example test in a single terminal:

```
ehubuux110> STDINOUT_Test stdinout.cfg
TTCN-3 Test Executor (single mode), version 1.7.pl4
Using configuration file: `stdinout.cfg'
Execution of control part in module STDINOUT_Test
started.
Test case HelloW2 started.
Hello, world!
Hello, TTCN-3!
Test case HelloW2 finished. Verdict: pass
Execution of control part in module STDINOUT_Test
finished.
Verdict statistics: 0 none (0.00 %), 1 pass (100.00 %), 0
inconc (0.00 %), 0 fail (0.00 %), 0 error (0.00 %).
Test execution summary: 1 test case was executed. Overall
verdict: pass
ehubuux110>
```

4.2 Parallel Execution Mode

When the Makefile includes “TTCN3_LIB = ttcn3-parallel” then the host controller terminal can be used for the input/output:

-- Main Controller Terminal --

```
ehubuux110> mctr_cli stdinout.cfg

*****************************************
* TTCN-3 Test Executor - Main Controller 2 *
* Version: 1.7.pl4 (R7E)                  *
*****************************************

Using configuration file: stdinout.cfg
MC@ehubuux110: Listening on TCP port 56550.
MC2>
```

-- Host Controller Terminal --

```
ehubuux110> STDINOUT_Test ehubuux110 56550
TTCN-3 Host Controller (parallel mode), version 1.7.pl4
```

-- Main Controller Terminal --

```
MC2> MC@ehubuux110: New HC connected from ehubuux110
[159.107.193.33]. ehubuux110: SunOS 5.8 on sun4u.
MC2> cmtc
MC@ehubuux110: Downloading configuration file to all HCs.
MC@ehubuux110: Configuration file was processed on all
HCs.
MC@ehubuux110: Creating MTC on host ehubuux110.
MC@ehubuux110: MTC is created.
MC2> smtc
Executing all items of [EXECUTE] section.
MC2> MTC@ehubuux110: Execution of control part in module
STDINOUT_Test started.
MTC@ehubuux110: Test case HelloW2 started.
```

-- Host Controller Terminal --

```
Hello, world!
Hello, TTCN-3!
```

-- Main Controller Terminal –

```
MTC@ehubuux110: Test case HelloW2 finished. Verdict: pass
MTC@ehubuux110: Execution of control part in module
STDINOUT_Test finished.
MC@ehubuux110: Test execution finished.
Execution of [EXECUTE] section finished.
MC2> emtc
MC@ehubuux110: Terminating MTC.
MTC@ehubuux110: Verdict statistics: 0 none (0.00 %), 1
pass (100.00 %), 0 inconc (0.00 %), 0 fail (0.00 %), 0
error (0.00 %).
MTC@ehubuux110: Test execution summary: 1 test case was
executed. Overall verdict: pass
MC@ehubuux110: MTC terminated.
MC2> exit
MC@ehubuux110: Shutting down session.
MC@ehubuux110: Shutdown complete.
```

4.3 Parallel Execution Mode with Autostart Script

When the Makefile includes “TTCN3_LIB = ttcn3-parallel” then the example shell script ttcn3_autostart.sh can also be used. This script needs the binary executable and the configuration file as parameters (NumHCs := 1 in the configuration file). For example it can be started as:

```
ttcn3_autostart.sh STDINOUT_Test stdinout.cfg
```

This script will open a new terminal which can be used for the input/output.