

Prepared (also subject responsible if other) ETH/RZD Jenő Balaskó +36 1 437 7760		No. 198 17-CNL113420 Uen		
Approved ETH/RZDC (Árpád Szakács)	Checked ETHGRY	Date 2007-01-10	Rev B	Reference GASK2

UDP Protocol Modules for TTCN-3 Toolset with TITAN, User Guide

Contents

1	Introduction.....	2
1.1	Revision history	2
1.2	About this Document	2
1.2.1	How to Read this Document.....	2
1.2.2	Presumed Knowledge	2
1.2.3	References	2
1.2.4	Abbreviations.....	2
1.2.5	Terminology	3
1.3	System Requirements	3
2	Protocol Modules.....	3
2.1	Overview	3
2.2	Installation	3
2.3	Configuration	3
3	Implementation specifics	4
4	Examples.....	4
4.1	UDP packet encoding and decoding	4

Prepared (also subject responsible if other) ETH/RZD Jenő Balaskó +36 1 437 7760		No. 198 17-CNL113420 Uen		
Approved ETH/RZDC (Árpád Szakács)	Checked ETHGRY	Date 2007-01-10	Rev B	Reference GASK2

1 Introduction

1.1 Revision history

Date	Rev	Characteristics	Prepared
2006-09-21	PA1	First draft version	ETHLZR
2006-11-14	A	Approved after review	ETHESI
2007-01-10	PB1	Updated for TITAN R7	ETHBAAT

1.2 About this Document

1.2.1 How to Read this Document

This is the User Guide for the UDP protocol module. The UDP protocol module is developed for the TTCN-3 Toolset with TITAN. This document should be read together with Product Revision Information [3] and Function Specification [4].

1.2.2 Presumed Knowledge

To use this protocol module the knowledge of the TTCN-3 language [1] is essential.

The specification of the UDP protocol is described in [5].

1.2.3 References

- [1] ETSI ES 201 873-1 v.3.1.1 (2005-06)
The Testing and Test Control Notation version 3. Part 1: Core Language
- [2] 1/1553-CRL 113 200 Uen
User Documentation for the TITAN TTCN-3 Test Executor
- [3] 109 21-CNL113 420-2
UDP Protocol Modules for TTCN-3 Toolset with TITAN, Product Revision Information
- [4] 155 17-CNL113 420
UDP Protocol Modules for TTCN-3 Toolset with TITAN, Function Specification
- [5] RFC 768 – User Datagram Protocol

1.2.4 Abbreviations

IPv4 Internet Protocol version 4

Prepared (also subject responsible if other) ETH/RZD Jenő Balaskó +36 1 437 7760		No. 198 17-CNL113420 Uen		
Approved ETH/RZDC (Árpád Szakács)	Checked ETHGRY	Date 2007-01-10	Rev B	Reference GASK2

IPv6 Internet Protocol version 6

RFC Request For Comments

TTCN-3 Testing and Test Control Notation version 3

UDP User Datagram Protocol

1.2.5 Terminology

No specific terminology is used.

1.3 System Requirements

Protocol modules are a set of TTCN-3 source code files that can be used as part of TTCN-3 test suites only. Hence, protocol modules alone do not put specific requirements on the system used. However in order to compile and execute a TTCN-3 test suite using the set of protocol modules the following system requirements must be satisfied:

- TITAN TTCN-3 Test Executor (1.7.pl0) or higher installed. For installation guide see [2]. Please note: This version of the protocol module is not compatible with TITAN releases earlier than R7A.

2 Protocol Modules

2.1 Overview

Protocol modules implement the messages structure of the related protocol in a formalized way, using the standard specification language TTCN-3. This allows defining of test data (templates) in the TTCN-3 language [1] and correctly encoding/decoding messages when executing test suites using the TITAN TTCN-3 test environment.

Protocol modules are using TITAN's RAW encoding attributes [2] and hence are usable with the TITAN test toolset only.

2.2 Installation

The set of protocol modules can be used in developing TTCN-3 test suites using any text editor. However to make the work more efficient a TTCN-3-enabled text editor is recommended (e.g. nedit, xemacs). Since the UDP protocol is used as a part of a TTCN-3 test suite, this requires TTCN-3 Test Executor be installed before the module can be compiled and executed together with other parts of the test suite. For more details on the installation of TTCN-3 Test Executor see the relevant section of [2].

2.3 Configuration

None.

Prepared (also subject responsible if other) ETH/RZD Jenő Balaskó +36 1 437 7760		No. 198 17-CNL113420 Uen		
Approved ETH/RZDC (Árpád Szakács)	Checked ETHGRY	Date 2007-01-10	Rev B	Reference GASK2

3 Implementation specifics

The `f_UDP_pseudo_header_enc()` can be used to encode the IP part of the UDP pseudo header. The parameter of the function is the UDP pseudo header. The return value is the encoded data.

The `f_UDP_checksum()` can be used to calculate the UDP checksum. The parameter of the function is the encoded UDP packet. The packet must contain the UDP checksum field and it must be zero. The return value is the calculated UDP checksum value. The length of the checksum is always 2 octets.

4 Examples

4.1 UDP packet encoding and decoding

The following example shows how a UDP packet can be encoded and decoded, when the UDP follows the IPv4 header and checksum calculation is enabled. The IPv6 case is exactly the same procedure.

```
var UDP_packet v_udp_packet;
var octetsring data;
var boolean udp_cksum_calc := true;

// Pseudo header in case the UDP follows an IPv4 header
template UDP_pseudo_header t_udp_pseudo_header_ipv4(LIN2_BO_LAST p_length) := {
  ipv4 := {
    srcaddr := '11223344'O,
    dstaddr := '11223345'O,
    zero := 0,
    proto := c_ip_proto_udp,
    plen := p_length
  }
}

// Encode the UDP packet
data := f_UDP_enc(v_udp_packet);

if (udp_cksum_calc)
{
  // calculate the UDP checksum value over the UDP pseudo header and the
  // encoded UDP packet
  udpcksum := f_UDP_checksum(f_UDP_pseudo_header_enc(valueof(
    t_udp_pseudo_header_ipv4(lengthof(data)))) & data);
  // Write the calculated checksum into the encoded UDP packet.
  // The checksum field is on the 7th and 8th octets.
  data[6] := udpcksum[0];
  data[7] := udpcksum[1];
}

// Decode the UDP packet
v_udp_packet := f_UDP_dec(data);
```