|  |  |
| --- | --- |
|  |  |

HTTP protocol module for TTCN-3 Toolset with TITAN, Function Description

# Abstract

The purpose of this document is to specify the functionality and usage of the HTTP protocol module.

Contents

[1 Introduction 2](#_Toc414473978)

[1.1 Revision history 2](#_Toc414473979)

[1.2 How to Read this Document 2](#_Toc414473980)

[1.3 Scope 2](#_Toc414473981)

[1.4 References 2](#_Toc414473982)

[1.5 Abbreviations 3](#_Toc414473983)

[1.6 Terminology 3](#_Toc414473984)

[1.7 System Requirements 3](#_Toc414473985)

[1.8 Installation 3](#_Toc414473986)

[1.9 Configuration 3](#_Toc414473987)

[2 Functional specification 4](#_Toc414473988)

[2.1 Protocol version implemented 4](#_Toc414473989)

[2.1.1 Implemented encoding/decoding and other related functions: 4](#_Toc414473990)

[2.1.2 Message length function 4](#_Toc414473991)

[2.2 Parser generation rules 4](#_Toc414473992)

# Introduction

## Revision history

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Rev | Characteristics | Prepared |
| 2015-01-13 | PA1 | Initial version | EESZSUS |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## How to Read this Document

This is the Function Specification for the set HTTP protocol module. HTTP protocol module is developed for the TTCN-3 Toolset with TITAN. This document should be read together with Product Revision Information [8].

## Scope

The purpose of this document is to specify the content of the HTTP protocol module. Basic knowledge of TTCN-3 [7] and TITAN TTCN-3 Test Executor [10] is valuable when reading this document.

## References

#### RFC 7230 Message Syntax and Routing

#### RFC 7231 Semantics and Content

#### RFC 7232 Conditional Requests

#### RFC 7233 Range Requests

#### RFC 7234 Caching

#### RFC 7235 Authentication

1. ETSI ES 201 873-1 v4.5.1 (2013-02)  
   The Testing and Test Control Notation version 3; Part 1: Core Language
2. 109 21-CNL 113 796-1 Uen  
   HTTP Protocol Module for TTCN-3 Toolset with TITAN, Product Revision Information
3. 2/198 17-CRL 113 200/5 Uen  
   Programmer’s Technical Reference for the TITAN TTCN-3 Test Executor
4. 1/198 17-CRL 113 200/5 Uen  
   User Guide for the TITAN TTCN-3 Test Executor

## Abbreviations

HTTP Hypertext Transfer Protocol

TTCN-3 Testing and Test Control Notation version 3

ETSI European Telecommunications Standards Institute

ITU-T International Telecommunication Union - Telecommunication Standardization Sector

## Terminology

No specific terminology is used.

## 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 version R8A (1.8.pl0) or higher installed. Please note: This version of the protocol module is not compatible with TITAN releases earlier than R8A.

## 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 HTTP 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 [9].

## Configuration

None.

# Functional specification

## Protocol version implemented

This set of protocol modules implements protocol messages and encode, decode functions of the HTTP protocol. The module is based on RFC 7230 (see [1]), RFC 7231 (see[2]), RFC 7232 (see[3]), RFC 7233 (see [4]), RFC 7234 (see[5]), RFC 7235 (see [6]). The following messages are implemented:

HTTP\_Message

Header fields that are not named can be listed in the header field called undefined\_header\_list as a name-value pair.

Header values can be given in the form of a list if the value is a list according to the standard. In an incoming message multiple header fields with the same name will be decoded as a list of lists.

### Implemented encoding/decoding and other related functions:

Name Type of formal parameters

ef\_HTTP\_Encode in HTTP\_Message pl\_pdu,

return octetstring

ef\_HTTP\_Decode in octetstring pl\_stream,

return HTTP\_Message

ef\_HTTPMessage\_len in octetstring stream

return integer

### Message length function

The f\_HTTPMessage\_len function returns the length of the HTTP message from an octetstring. If the length cannot be determined it returns the value -1.

## Parser generation rules

In order to generate the .c and .h files from .y and .l the following Makefile rules should be used:

HTTP\_parse\_.tab.c HTTP\_parse\_.tab.h: HTTP\_parse.y

bison -t -dv -p HTTP\_parse\_ -b HTTP\_parse\_ $<

lex.HTTP\_parse\_.c: HTTP\_parse.l

flex -Cr -8 -Bvpp -P HTTP\_parse\_ $<

The .h and .c parser files should be generated during the protocol module development. Only the pregenerated files are needed for test case development and test execution.