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

Eszter Susánszky

Version 1551-CNL 113 796, Rev. A, 2015-03-18

Table of Contents

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.

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 [9] is valuable when reading this document.

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 [8].

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, RFC 7231, RFC 7232, RFC 7233, RFC 7234, RFC 7235). 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

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.

Terminology

No specific terminology is used.

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

References

[1] RFC 7230
Message Syntax and Routing

[2] RFC 7231
Semantics and Content

[3] RFC 7232
Conditional Requests

[4] RFC 7233
Range Requests

[5] RFC 7234
Caching

[6] RFC 7235
Authentication

[7] ETSI ES 201 873-1 v4.5.1 (2013-02)
The Testing and Test Control Notation version 3; Part 1: Core Language

[8] Programmer's Technical Reference for the TITAN TTCN-3 Test Executor

[9] User Guide for the TITAN TTCN-3 Test Executor