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

EPTF AppLib HTTP for TitanSim, Function Specification

Contents

[1 Introduction 4](#_Toc310400818)

[1.1 Revision history 4](#_Toc310400819)

[1.2 How to Read this Document 4](#_Toc310400820)

[1.3 Scope 4](#_Toc310400821)

[1.4 References 4](#_Toc310400822)

[1.5 Typographical conventions 5](#_Toc310400823)

[1.6 Abbreviations 5](#_Toc310400824)

[1.7 Terminology 5](#_Toc310400825)

[2 The implemented modules 6](#_Toc310400826)

[2.1 General 6](#_Toc310400827)

[3 Functional specification 6](#_Toc310400828)

[3.1 Protocol version implemented 6](#_Toc310400829)

[3.1.1 Implemented Functionality 6](#_Toc310400830)

[3.1.2 Protocol Modifications/Deviations 8](#_Toc310400831)

[3.2 External templates 8](#_Toc310400832)

[3.3 Encoding/Decoding and Other Related Functions 8](#_Toc310400833)

[3.3.1 Header – internal encoding 9](#_Toc310400834)

[3.3.2 Header – raw header 9](#_Toc310400835)

[3.3.3 Header – external template 9](#_Toc310400836)

[3.3.4 Body – internal encoding and external template 9](#_Toc310400837)

[3.4 Test steps 9](#_Toc310400838)

[3.4.1 f\_EPTF\_HTTP\_step\_sendRequest 10](#_Toc310400839)

[3.4.2 f\_EPTF\_HTTP\_step\_handleTimeout 10](#_Toc310400840)

[3.4.3 f\_EPTF\_HTTP\_step\_processResponse 10](#_Toc310400841)

[3.4.4 f\_EPTF\_HTTP\_step\_closePortOfUser 10](#_Toc310400842)

[3.4.5 f\_EPTF\_HTTP\_step\_freePortOfUser 10](#_Toc310400843)

[3.4.6 f\_EPTF\_HTTP\_step\_sendConnect 10](#_Toc310400844)

[3.4.7 f\_EPTF\_HTTP\_step\_sendClose 10](#_Toc310400845)

[3.4.8 f\_EPTF\_HTTP\_step\_sendHalfClose 10](#_Toc310400846)

[3.4.9 f\_EPTF\_HTTP\_step\_setEntityContextHTTPMethodCONNECT 11](#_Toc310400847)

[3.4.10 f\_EPTF\_HTTP\_step\_setEntityContextHTTPMethodDELETE 11](#_Toc310400848)

[3.4.11 f\_EPTF\_HTTP\_step\_setEntityContextHTTPMethodGET 11](#_Toc310400849)

[3.4.12 f\_EPTF\_HTTP\_step\_setEntityContextHTTPMethodHEAD 11](#_Toc310400850)

[3.4.13 f\_EPTF\_HTTP\_step\_setEntityContextHTTPMethodOPTIONS 11](#_Toc310400851)

[3.4.14 f\_EPTF\_HTTP\_step\_setEntityContextHTTPMethodPOST 11](#_Toc310400852)

[3.4.15 f\_EPTF\_HTTP\_step\_setEntityContextHTTPMethodPUT 11](#_Toc310400853)

[3.4.16 f\_EPTF\_HTTP\_step\_setEntityContextHTTPMethodTRACE 11](#_Toc310400854)

[3.4.17 f\_EPTF\_HTTP\_step\_handleRequest 11](#_Toc310400855)

[3.4.18 f\_EPTF\_HTTP\_step\_sendResponse 11](#_Toc310400856)

[3.4.19 f\_EPTF\_HTTP\_step\_closePort 11](#_Toc310400857)

[3.4.20 f\_EPTF\_HTTP\_step\_cleanUp 11](#_Toc310400858)

[3.5 STEP Constants 12](#_Toc310400859)

[3.5.1 c\_EPTF\_HTTP\_stepIdx\_sendRequest 12](#_Toc310400860)

[3.5.2 c\_EPTF\_HTTP\_stepIdx\_handleTimeout 12](#_Toc310400861)

[3.5.3 c\_EPTF\_HTTP\_stepIdx\_responseReceived 12](#_Toc310400862)

[3.5.4 c\_EPTF\_HTTP\_stepIdx\_responseCharReceived 12](#_Toc310400863)

[3.5.5 c\_EPTF\_HTTP\_stepIdx\_requestCharReceived 12](#_Toc310400864)

[3.5.6 c\_EPTF\_HTTP\_stepIdx\_responseOctReceived 12](#_Toc310400865)

[3.5.7 c\_EPTF\_HTTP\_stepIdx\_requestCharReceived 12](#_Toc310400866)

[3.5.8 c\_EPTF\_HTTP\_stepIdx\_openConnection 12](#_Toc310400867)

[3.5.9 c\_EPTF\_HTTP\_stepIdx\_closeConnection 13](#_Toc310400868)

[3.5.10 c\_EPTF\_HTTP\_stepIdx\_halfCloseConnection 13](#_Toc310400869)

[3.5.11 c\_EPTF\_HTTP\_stepIdx\_closePortOfUser 13](#_Toc310400870)

[3.5.12 c\_EPTF\_HTTP\_stepIdx\_freePortOfUser 13](#_Toc310400871)

[3.5.13 c\_EPTF\_HTTP\_stepIdx\_setEntityContextHTTPMethodCONNECT 13](#_Toc310400872)

[3.5.14 c\_EPTF\_HTTP\_stepIdx\_setEntityContextHTTPMethodDELETE 13](#_Toc310400873)

[3.5.15 c\_EPTF\_HTTP\_stepIdx\_setEntityContextHTTPMethodGET 13](#_Toc310400874)

[3.5.16 c\_EPTF\_HTTP\_stepIdx\_setEntityContextHTTPMethodHEAD 13](#_Toc310400875)

[3.5.17 c\_EPTF\_HTTP\_stepIdx\_setEntityContextHTTPMethodOPTIONS 13](#_Toc310400876)

[3.5.18 c\_EPTF\_HTTP\_stepIdx\_setEntityContextHTTPMethodPOST 14](#_Toc310400877)

[3.5.19 c\_EPTF\_HTTP\_stepIdx\_setEntityContextHTTPMethodPUT 14](#_Toc310400878)

[3.5.20 c\_EPTF\_HTTP\_stepIdx\_setEntityContextHTTPMethodTRACE 14](#_Toc310400879)

[3.5.21 c\_EPTF\_HTTP\_stepIdx\_handleRequest 14](#_Toc310400880)

[3.5.22 c\_EPTF\_HTTP\_stepIdx\_sendResponse 14](#_Toc310400881)

[3.5.23 c\_EPTF\_HTTP\_stepIdx\_closePort 14](#_Toc310400882)

[3.5.24 c\_EPTF\_HTTP\_stepIdx\_cleanUpContext 14](#_Toc310400883)

[3.6 Inputs 14](#_Toc310400884)

[3.6.1 c\_EPTF\_HTTP\_inputIdx\_connectionOpened 14](#_Toc310400885)

[3.6.2 c\_EPTF\_HTTP\_inputIdx\_connectionClosed 14](#_Toc310400886)

[3.6.3 c\_EPTF\_HTTP\_inputIdx\_errorReceived 14](#_Toc310400887)

[3.6.4 c\_EPTF\_HTTP\_inputIdx\_OKReceived 15](#_Toc310400888)

[3.6.5 c\_EPTF\_HTTP\_inputIdx\_response100Continue 15](#_Toc310400889)

[3.6.6 c\_EPTF\_HTTP\_inputIdx\_response101SwitchingProtocols 15](#_Toc310400890)

[3.6.7 c\_EPTF\_HTTP\_inputIdx\_response200OK 15](#_Toc310400891)

[3.6.8 c\_EPTF\_HTTP\_inputIdx\_response202Accepted 15](#_Toc310400892)

[3.6.9 c\_EPTF\_HTTP\_inputIdx\_response203NonAuthorativeInformation 15](#_Toc310400893)

[3.6.10 c\_EPTF\_HTTP\_inputIdx\_response204NoContent 15](#_Toc310400894)

[3.6.11 c\_EPTF\_HTTP\_inputIdx\_response205ResetContent 15](#_Toc310400895)

[3.6.12 c\_EPTF\_HTTP\_inputIdx\_response206PartialContent 16](#_Toc310400896)

[3.6.13 c\_EPTF\_HTTP\_inputIdx\_response300MultipleChoices 16](#_Toc310400897)

[3.6.14 c\_EPTF\_HTTP\_inputIdx\_response301MovedPermanently 16](#_Toc310400898)

[3.6.15 c\_EPTF\_HTTP\_inputIdx\_response302Found 16](#_Toc310400899)

[3.6.16 c\_EPTF\_HTTP\_inputIdx\_response303SeeOther 16](#_Toc310400900)

[3.6.17 c\_EPTF\_HTTP\_inputIdx\_response304NotModified 16](#_Toc310400901)

[3.6.18 c\_EPTF\_HTTP\_inputIdx\_response305UseProxy 16](#_Toc310400902)

[3.6.19 c\_EPTF\_HTTP\_inputIdx\_response306Unused 17](#_Toc310400903)

[3.6.20 c\_EPTF\_HTTP\_inputIdx\_response307TemporaryRedirect 17](#_Toc310400904)

[3.6.21 c\_EPTF\_HTTP\_inputIdx\_response400BadRequest 17](#_Toc310400905)

[3.6.22 c\_EPTF\_HTTP\_inputIdx\_response401Unauthorized 17](#_Toc310400906)

[3.6.23 c\_EPTF\_HTTP\_inputIdx\_response402PaymentRequired 17](#_Toc310400907)

[3.6.24 c\_EPTF\_HTTP\_inputIdx\_response403Forbidden 17](#_Toc310400908)

[3.6.25 c\_EPTF\_HTTP\_inputIdx\_response404NotFound 17](#_Toc310400909)

[3.6.26 c\_EPTF\_HTTP\_inputIdx\_response405MethodNotAllowed 17](#_Toc310400910)

[3.6.27 c\_EPTF\_HTTP\_inputIdx\_response406NotAcceptable 18](#_Toc310400911)

[3.6.28 c\_EPTF\_HTTP\_inputIdx\_response407ProxyAuthenticationRequired 18](#_Toc310400912)

[3.6.29 c\_EPTF\_HTTP\_inputIdx\_response408RequestTimeout 18](#_Toc310400913)

[3.6.30 c\_EPTF\_HTTP\_inputIdx\_response409Conflict 18](#_Toc310400914)

[3.6.31 c\_EPTF\_HTTP\_inputIdx\_response410Gone 18](#_Toc310400915)

[3.6.32 c\_EPTF\_HTTP\_inputIdx\_response411LengthRequired 18](#_Toc310400916)

[3.6.33 c\_EPTF\_HTTP\_inputIdx\_response412PreconditionFailed 18](#_Toc310400917)

[3.6.34 c\_EPTF\_HTTP\_inputIdx\_response413RequestEntityTooLarge 19](#_Toc310400918)

[3.6.35 c\_EPTF\_HTTP\_inputIdx\_response414RequestURITooLong 19](#_Toc310400919)

[3.6.36 c\_EPTF\_HTTP\_inputIdx\_response415UnsupportedMediaType 19](#_Toc310400920)

[3.6.37 c\_EPTF\_HTTP\_inputIdx\_response416RequestedRangeNotSatisfiable 19](#_Toc310400921)

[3.6.38 c\_EPTF\_HTTP\_inputIdx\_response417ExpectationFailed 19](#_Toc310400922)

[3.6.39 c\_EPTF\_HTTP\_inputIdx\_response500InternalServerError 19](#_Toc310400923)

[3.6.40 c\_EPTF\_HTTP\_inputIdx\_response501NotImplemented 19](#_Toc310400924)

[3.6.41 c\_EPTF\_HTTP\_inputIdx\_response502BadGateway 19](#_Toc310400925)

[3.6.42 c\_EPTF\_HTTP\_inputIdx\_response503ServiceUnavailable 20](#_Toc310400926)

[3.6.43 c\_EPTF\_HTTP\_inputIdx\_response504GatewayTimeout 20](#_Toc310400927)

[3.6.44 c\_EPTF\_HTTP\_inputIdx\_response505HTTPVersionNotSupported 20](#_Toc310400928)

[3.6.45 c\_EPTF\_HTTP\_inputIdx\_response100class 20](#_Toc310400929)

[3.6.46 c\_EPTF\_HTTP\_inputIdx\_response200class 20](#_Toc310400930)

[3.6.47 c\_EPTF\_HTTP\_inputIdx\_response300class 20](#_Toc310400931)

[3.6.48 c\_EPTF\_HTTP\_inputIdx\_response400class 20](#_Toc310400932)

[3.6.49 c\_EPTF\_HTTP\_inputIdx\_response500class 21](#_Toc310400933)

[3.6.50 c\_EPTF\_HTTP\_inputIdx\_responseSocketError 21](#_Toc310400934)

[3.7 FSMs 21](#_Toc310400935)

[3.8 Statistics 21](#_Toc310400936)

[3.8.1 c\_EPTF\_HTTP\_Stat\_nofIncRequests 21](#_Toc310400937)

[3.8.2 c\_EPTF\_HTTP\_Stat\_nofIncOptionMessages 21](#_Toc310400938)

[3.8.3 c\_EPTF\_HTTP\_Stat\_nofOutOptionMessages 21](#_Toc310400939)

[3.8.4 c\_EPTF\_HTTP\_Stat\_nofIncGetMessages 21](#_Toc310400940)

[3.8.5 c\_EPTF\_HTTP\_Stat\_nofOutRequests 21](#_Toc310400941)

[3.8.6 c\_EPTF\_HTTP\_Stat\_nofOutGetMessages 21](#_Toc310400942)

[3.8.7 c\_EPTF\_HTTP\_Stat\_nofIncHeadMessages 21](#_Toc310400943)

[3.8.8 c\_EPTF\_HTTP\_Stat\_nofOutHeadMessages 22](#_Toc310400944)

[3.8.9 c\_EPTF\_HTTP\_Stat\_nofIncPortMessages 22](#_Toc310400945)

[3.8.10 c\_EPTF\_HTTP\_Stat\_nofOutPostMessages 22](#_Toc310400946)

[3.8.11 c\_EPTF\_HTTP\_Stat\_nofIncPutMessages 22](#_Toc310400947)

[3.8.12 c\_EPTF\_HTTP\_Stat\_nofOutPutMessages 22](#_Toc310400948)

[3.8.13 c\_EPTF\_HTTP\_Stat\_nofIncDeleteMessages 22](#_Toc310400949)

[3.8.14 c\_EPTF\_HTTP\_Stat\_nofOutDeleteMessages 22](#_Toc310400950)

[3.8.15 c\_EPTF\_HTTP\_Stat\_nofIncTraceMessages 22](#_Toc310400951)

[3.8.16 c\_EPTF\_HTTP\_Stat\_nofOutTraceMessages 22](#_Toc310400952)

[3.8.17 c\_EPTF\_HTTP\_Stat\_nofIncConnectMessages 22](#_Toc310400953)

[3.8.18 c\_EPTF\_HTTP\_Stat\_nofOutConnectMessages 22](#_Toc310400954)

[3.8.19 c\_EPTF\_HTTP\_Stat\_nofIncResponses 22](#_Toc310400955)

[3.8.20 c\_EPTF\_HTTP\_Stat\_nofIncXXXStatusCode 22](#_Toc310400956)

[3.8.21 c\_EPTF\_HTTP\_Stat\_nofOutResponses 23](#_Toc310400957)

[3.8.22 c\_EPTF\_HTTP\_Stat\_nofOutXXXStatusCode 23](#_Toc310400958)

# Introduction

## Revision history

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Rev | Characteristics | Prepared |
| 2009-01-08 | PA1 | First draft version | EAKOPER |
| 2009-01-19 | PA2 | Updated after review | EAKOPER |
| 2009-03-10 | PA3 | External template support added | EPTEDIM |
| 2009-04-28 | PA4 | Updated after bug fixing | EBENMOL |
| 2009-08-10 | PB1 | Added server functionality;  Added server side Digest authentication | EGERGFT |
| 2011-10-25 | PC1 | Functionality in the remote transport to get the target LGen component based on the URI in the request has been added. | ETHECS |

## How to Read this Document

This is the Function Specification for the HTTP Application Library. The modules are developed by the TTCN-3 Toolset with TITAN. This document shall be read together with Product Revision Information [5] and the User Guide [6].

## Scope

The purpose of this document is to specify the content of the HTTP Application Library modules.

## References

1. [RFC 1945](http://www.ietf.org/rfc/rfc1945.txt)  
   Hypertext Transfer Protocol – HTTP/1.0

1. [RFC 2616](http://www.ietf.org/rfc/rfc2616.txt)  
   Hypertext Transfer Protocol – HTTP/1.1
2. [RFC 2617](http://tools.ietf.org/html/rfc2617)  
   HTTP Authentication: Basic and Digest Access Authentication
3. ETSI ES 201 873-1 v.3.1.1  
   The Testing and Test Control Notation version 3. Part 1: Core Language, July 2006
4. 109 21-CNL 113 618-1  
   EPTF AppLib HTTP for TitanSim, Product Revision Information
5. 198 17/CNL 113 618 Uen  
   EPTF AppLib HTTP for TitanSim, User Guide
6. 1/1553-CRL 113 200 Uen  
   User Documentation for the TITAN TTCN-3 Test Executor
7. 155 17-CNL 113 512 Uen  
   EPTF Core Library for TTCN-3 toolset with TITAN, Function Specification.

## Typographical conventions

Important concepts are denoted by *italic* font wherever they are first used in the given context.

## Abbreviations

EPTF Ericsson Performance Test Framework

FSM Finite State Machine

HTTP Hypertext Transfer Protocol

SUT System Under Test

TTCN-3 Testing and Test Control Notation version 3

## Terminology

*Port*

A port represents connections (incoming/outgoing) or listening sockets.

*Port group*

A port group is a set of ports with a unique group identifier. A port group contains one or more ports.

*TitanSim Core (Load) Library(CLL)*

The part of the TitanSim software that is totally project independent. (I.e., which is not protocol-, or application-dependent). The TitanSim CLL is to be supplied and supported by the TCC organization. Any TitanSim CLL development is to be funded centrally by Ericsson.

*TitanSim Appliaction Library*

The application-specific part of the TitanSim software.It provides load generation functionalities belonging to specific protocols or products.

# The implemented modules

## General

The following modules are elements of an Application Library, part of the EPTF [8]. With these modules we can create a user defined HTTP load generator. Library functions can be easily replaced by user defined ones, so HTTP Application Library can be extended for more specific use.

# Functional specification

## Protocol version implemented

This application library implements basic protocol behaviour of the HTTP protocol based on [1] and [2]. Modifications/deviations related to the protocol specification

### Implemented Functionality

The application library is able to send HTTP queries or responses, maintains a number of built-in statistics and handles different transport modes.

#### HTTP Context

The application library provides a HTTP context database which is used to store user related information. Common data for requests and responses:

* + Major version number
  + Minor version number
  + header lines: a sequence of HTTP header name and value pairs
  + connection id: the unique id of an HTTP connection object
  + message body: HTTP message body (it can be binary body)

Data for requests:

* + method: HTTP method (e.g.: POST, GET etc)
  + URI: HTTP request URI
  + authentication details: authentication details of the user

Data for responses:

* + statusCode: status code of the response
  + statusText: status text of the response

Above information is used to build an HTTP message.

#### HTTP Statistics

Counter statistics are provided for incoming and outgoing messages. The number of incoming and outgoing messages is counted based on HTTP method and HTTP status code properties.

#### HTTP Transport

A port group is a set of ports with a unique id. A port group contains one ore more ports. Ports are representing connections and listen sockets. The user can perform an operation on the port group using its unique id.

HTTP messages are sent and received trough the HTTP transport layer. The transport layer is able to maintain a number of HTTP connection objects. These connection objects are the ports.

A port group can work either in a client or a server mode.

**Client mode**

By means of a client port group, requests can be sent. The client port group handles the incoming responses.

Port groups have instant connection open, instant connection closed and use SSL properties. All of the ports in the group will have the same property setting.

Instant connection open means that before sending on a port, a connection open operation will be attempted on the port.

Instant connection close means that after receiving on a port, a connection close operation will be attempted on the port.

Use SSL is reserved for future use to implement security.

**Server mode**

By means of a server port group, a listen socket can be opened and responses can be sent to incoming requests. The server port group also handles incoming connections.

#### Persistent connections

Server port groups support the persistent connection handling and the pipelining (see [2]). A buffer is maintained for the responses so the responses will be sent in the order that the requests were received. The size of the buffer is a configuration parameter.

### Protocol Modifications/Deviations

None.

## External templates

The application library defines three external template types.

* Template for request headers (CUSTOM\_HEADER) with one mandatory parameter *method* and several optional parameters: *requestURI, versionMajor, versionMinor, customHeader, contentLength.*
* Template for response headers (CUSTOM\_HEADER\_RESPONSE) with the optional parameters: *status, versionMajor, versionMinor, contentLength*
* Template for bodies (CUSTOM\_BODY) with no parameters. That is, template text is copied verbatim into the HTTP message.

Sample request header template:

$(method) $(requestURI) HTTP/$(versionMajor).$(versionMinor)

DemoExternalHeader : test

Content-Length : $(contentLength)

$(customHeader)

* Sample response header:

HTTP/$(versionMajor).$(versionMinor)

$(status)

DemoExternalHeader : test

Server: local

Content-Length : $(contentLength)

* For proper encoding the template files shall use the standard Unix line separator: “\n”.

## Encoding/Decoding and Other Related Functions

In the application library three different encoding scenario has been implemented for header encoding, and two for body encoding.

### Header – internal encoding

This is the base case when no external templates are defined and no EPTF\_HTTP\_rawHeaderContentGetter\_FT function set. Message encoding is implemented in the HTTP test port (CNL 113 531). Message parameters are retrieved from the HTTP context.

### Header – raw header

No encoding occurs when no external templates are defined and EPTF\_HTTP\_rawHeaderContentGetter\_FT function is set. In this case the upper layer provides the encoded header to the Applib which in turn is copied verbatim to the outgoing HTTP message.

### Header – external template

External templates have the highest priority. If external templates are defined then external templates will be used whether any content getter function is set or not. The HTTP context provides the actual parameters to the external template instance.

#### Request template

In this context the *customHeader* parameter will contain the header lines for the request template defined in the HTTP context.

#### Response template

For adding further optional parameters to the response template type, the LGenBase feature provides the following function: f\_EPTF\_LGenBase\_extendTemplateType. The number of additional template parameters must be set by means of the f\_EPTF\_HTTP\_addOptTempParamNumber function in the AppLib.

The parameters values that should be assigned to the optional parameters added by the user must be set in the HTTP context by means of the f\_EPTF\_HTTP\_setOptTempParam function.

### Body – internal encoding and external template

If external templates are defined the content of the external template will be used as the body of the HTTP message. Otherwise the body is determined by the getter function set in the HTTP context.

## Test steps

Test steps are used to apply the application library’s functionality in the finite state machine (FSM) of a specific traffic case. The user can extend basic test steps to achieve a more specific behaviour.

### f\_EPTF\_HTTP\_step\_sendRequest

This step sends an HTTP request based on HTTP context database and updates HTTP statistics. The query is assembled from the context database element which belongs to the entity to which the step has been applied.

### f\_EPTF\_HTTP\_step\_handleTimeout

This step handles timeout after sending an HTTP request. The function currently does nothing. The user should set its own timeout handling step.

### f\_EPTF\_HTTP\_step\_processResponse

This step handles reception of HTTP response after sending an HTTP request. The function currently does nothing. The user should set its own message reception handling step.

### f\_EPTF\_HTTP\_step\_closePortOfUser

If a user sends on a port or on a port of a port group, then the port is associated to the user. This case the user can close its associated port by applying the current test step.

### f\_EPTF\_HTTP\_step\_freePortOfUser

If a user sends on a port or on a port of a port group, then the port is associated to the user. This case the user can make its associated port free for sending (without closing this) by applying the current test step.

### f\_EPTF\_HTTP\_step\_sendConnect

This step sends an HTTP connect request towards the HTTP connection object associated to the entity to which the step has been applied.

Note that if the connection id identifies a port group, then the request is applied to all ports in the port group.

### f\_EPTF\_HTTP\_step\_sendClose

This step sends an HTTP close request towards the HTTP connection object associated to the entity to which the step has been applied.

Note that if the connection id identifies a port group, then the request is applied to all ports in the port group.

### f\_EPTF\_HTTP\_step\_sendHalfClose

This step sends an HTTP half close request towards the HTTP connection object associated to the entity to which the step has been applied.

Note that if the connection id identifies a port group, then the request is applied to all ports in the port group.

### f\_EPTF\_HTTP\_step\_setEntityContextHTTPMethodCONNECT

This test step sets up the entity context to the Connect HTTP Method.

### f\_EPTF\_HTTP\_step\_setEntityContextHTTPMethodDELETE

This test step sets up the entity context to the Delete HTTP Method.

### f\_EPTF\_HTTP\_step\_setEntityContextHTTPMethodGET

This test step sets up the entity context to the Get HTTP Method.

### f\_EPTF\_HTTP\_step\_setEntityContextHTTPMethodHEAD

This test step sets up the entity context to the Head HTTP Method.

### f\_EPTF\_HTTP\_step\_setEntityContextHTTPMethodOPTIONS

This test step sets up the entity context to the Options HTTP Method.

### f\_EPTF\_HTTP\_step\_setEntityContextHTTPMethodPOST

This test step sets up the entity context to the Ppost HTTP Method.

### f\_EPTF\_HTTP\_step\_setEntityContextHTTPMethodPUT

This test step sets up the entity context to the Put HTTP Method.

### f\_EPTF\_HTTP\_step\_setEntityContextHTTPMethodTRACE

This test step sets up the entity context to the Trace HTTP Method.

### f\_EPTF\_HTTP\_step\_handleRequest

This test step must be called to handle an incoming request.

### f\_EPTF\_HTTP\_step\_sendResponse

This test step sends a response to a request.

### f\_EPTF\_HTTP\_step\_closePort

This test step closes a port in the server port group.

### f\_EPTF\_HTTP\_step\_cleanUp

This test step cleans up the entity context.

## STEP Constants

The following step constants are defined in the application library:

### c\_EPTF\_HTTP\_stepIdx\_sendRequest

Scope: entity

Description: trigger for HTTP send request

### c\_EPTF\_HTTP\_stepIdx\_handleTimeout

Scope: entity

Description: trigger for HTTP timeout event

### c\_EPTF\_HTTP\_stepIdx\_responseReceived

Scope: entity

Description: trigger for HTTP response message reception event

### c\_EPTF\_HTTP\_stepIdx\_responseCharReceived

Scope: entity

Description: trigger for HTTP response message reception with charstring body event

### c\_EPTF\_HTTP\_stepIdx\_requestCharReceived

Scope: entity

Description: trigger for HTTP request message reception with charstring body event

### c\_EPTF\_HTTP\_stepIdx\_responseOctReceived

Scope: entity

Description: trigger for HTTP response message reception with binary body event

### c\_EPTF\_HTTP\_stepIdx\_requestCharReceived

Scope: entity

Description: trigger for HTTP request message reception with binary body event

### c\_EPTF\_HTTP\_stepIdx\_openConnection

Scope: entity

Description: triggers an HTTP open connection event

### c\_EPTF\_HTTP\_stepIdx\_closeConnection

Scope: entity

Description: trigger for an HTTP close connection event

### c\_EPTF\_HTTP\_stepIdx\_halfCloseConnection

Scope: entity

Description: trigger for an HTTP half close connection event

### c\_EPTF\_HTTP\_stepIdx\_closePortOfUser

Scope: entity

Description: trigger for an HTTP close port of user event

### c\_EPTF\_HTTP\_stepIdx\_freePortOfUser

Scope: entity

Description: trigger for an HTTP free port of user event

### c\_EPTF\_HTTP\_stepIdx\_setEntityContextHTTPMethodCONNECT

Scope: entity

Description: trigger for set up the HTTP Connect Method.

### c\_EPTF\_HTTP\_stepIdx\_setEntityContextHTTPMethodDELETE

Scope: entity

Description: trigger for set up the HTTP Delete Method.

### c\_EPTF\_HTTP\_stepIdx\_setEntityContextHTTPMethodGET

Scope: entity

Description: trigger for set up the HTTP Get Method.

### c\_EPTF\_HTTP\_stepIdx\_setEntityContextHTTPMethodHEAD

Scope: entity

Description: trigger for set up the HTTP Head Method.

### c\_EPTF\_HTTP\_stepIdx\_setEntityContextHTTPMethodOPTIONS

Scope: entity

Description: trigger for set up the HTTP Options Method.

### c\_EPTF\_HTTP\_stepIdx\_setEntityContextHTTPMethodPOST

Scope: entity

Description: trigger for set up the HTTP Post Method.

### c\_EPTF\_HTTP\_stepIdx\_setEntityContextHTTPMethodPUT

Scope: entity

Description: trigger for set up the HTTP Put Method.

### c\_EPTF\_HTTP\_stepIdx\_setEntityContextHTTPMethodTRACE

Scope: entity

Description: trigger for set up the HTTP Trace Method.

### c\_EPTF\_HTTP\_stepIdx\_handleRequest

Index constant for f\_EPTF\_HTTP\_step\_handleRequest.

### c\_EPTF\_HTTP\_stepIdx\_sendResponse

Index constant for f\_EPTF\_HTTP\_step\_sendResponse.

### c\_EPTF\_HTTP\_stepIdx\_closePort

Index constant for f\_EPTF\_HTTP\_step\_closePort.

### c\_EPTF\_HTTP\_stepIdx\_cleanUpContext

Index constant for f\_EPTF\_HTTP\_step\_cleanUpContext.

## Inputs

### c\_EPTF\_HTTP\_inputIdx\_connectionOpened

Scope: entity

Description: trigger for an HTTP connection opened event

### c\_EPTF\_HTTP\_inputIdx\_connectionClosed

Scope: entity

Description: trigger for an HTTP connection closed event

### c\_EPTF\_HTTP\_inputIdx\_errorReceived

Scope: entity

Description: trigger for an HTTP error reception event

### c\_EPTF\_HTTP\_inputIdx\_OKReceived

Scope: entity

Description: trigger for an HTTP OK received event

### c\_EPTF\_HTTP\_inputIdx\_response100Continue

Scope: entity

Description: trigger for an HTTP message reception event with status code 100

### c\_EPTF\_HTTP\_inputIdx\_response101SwitchingProtocols

Scope: entity

Description: trigger for an HTTP message reception event with status code 101

### c\_EPTF\_HTTP\_inputIdx\_response200OK

Scope: entity

Description: trigger for an HTTP message reception event with status code 200

### c\_EPTF\_HTTP\_inputIdx\_response202Accepted

Scope: entity

Description: trigger for an HTTP message reception event with status code 202

### c\_EPTF\_HTTP\_inputIdx\_response203NonAuthorativeInformation

Scope: entity

Description: trigger for an HTTP message reception event with status code 203

### c\_EPTF\_HTTP\_inputIdx\_response204NoContent

Scope: entity

Description: trigger for an HTTP message reception event with status code 204

### c\_EPTF\_HTTP\_inputIdx\_response205ResetContent

Scope: entity

Description: trigger for an HTTP message reception event with status code 205

### c\_EPTF\_HTTP\_inputIdx\_response206PartialContent

Scope: entity

Description: trigger for an HTTP message reception event with status code 206

### c\_EPTF\_HTTP\_inputIdx\_response300MultipleChoices

Scope: entity

Description: trigger for an HTTP message reception event with status code 300

### c\_EPTF\_HTTP\_inputIdx\_response301MovedPermanently

Scope: entity

Description: trigger for an HTTP message reception event with status code 301

### c\_EPTF\_HTTP\_inputIdx\_response302Found

Scope: entity

Description: trigger for an HTTP message reception event with status code 302

### c\_EPTF\_HTTP\_inputIdx\_response303SeeOther

Scope: entity

Description: trigger for an HTTP message reception event with status code 303

### c\_EPTF\_HTTP\_inputIdx\_response304NotModified

Scope: entity

Description: trigger for an HTTP message reception event with status code 304

### c\_EPTF\_HTTP\_inputIdx\_response305UseProxy

Scope: entity

Description: trigger for an HTTP message reception event with status code 305

### c\_EPTF\_HTTP\_inputIdx\_response306Unused

Scope: entity

Description: trigger for an HTTP message reception event with status code 306

### c\_EPTF\_HTTP\_inputIdx\_response307TemporaryRedirect

Scope: entity

Description: trigger for an HTTP message reception event with status code 307

### c\_EPTF\_HTTP\_inputIdx\_response400BadRequest

Scope: entity

Description: trigger for an HTTP message reception event with status code 400

### c\_EPTF\_HTTP\_inputIdx\_response401Unauthorized

Scope: entity

Description: trigger for an HTTP message reception event with status code 401

### c\_EPTF\_HTTP\_inputIdx\_response402PaymentRequired

Scope: entity

Description: trigger for an HTTP message reception event with status code 402

### c\_EPTF\_HTTP\_inputIdx\_response403Forbidden

Scope: entity

Description: trigger for an HTTP message reception event with status code 403

### c\_EPTF\_HTTP\_inputIdx\_response404NotFound

Scope: entity

Description: trigger for an HTTP message reception event with status code 404

### c\_EPTF\_HTTP\_inputIdx\_response405MethodNotAllowed

Scope: entity

Description: trigger for an HTTP message reception event with status code 405

### c\_EPTF\_HTTP\_inputIdx\_response406NotAcceptable

Scope: entity

Description: trigger for an HTTP message reception event with status code 406

### c\_EPTF\_HTTP\_inputIdx\_response407ProxyAuthenticationRequired

Scope: entity

Description: trigger for an HTTP message reception event with status code 407

### c\_EPTF\_HTTP\_inputIdx\_response408RequestTimeout

Scope: entity

Description: trigger for an HTTP message reception event with status code 408

### c\_EPTF\_HTTP\_inputIdx\_response409Conflict

Scope: entity

Description: trigger for an HTTP message reception event with status code 409

### c\_EPTF\_HTTP\_inputIdx\_response410Gone

Scope: entity

Description: trigger for an HTTP message reception event with status code 410

### c\_EPTF\_HTTP\_inputIdx\_response411LengthRequired

Scope: entity

Description: trigger for an HTTP message reception event with status code 411

### c\_EPTF\_HTTP\_inputIdx\_response412PreconditionFailed

Scope: entity

Description: trigger for an HTTP message reception event with status code 412

### c\_EPTF\_HTTP\_inputIdx\_response413RequestEntityTooLarge

Scope: entity

Description: trigger for an HTTP message reception event with status code 413

### c\_EPTF\_HTTP\_inputIdx\_response414RequestURITooLong

Scope: entity

Description: trigger for an HTTP message reception event with status code 414

### c\_EPTF\_HTTP\_inputIdx\_response415UnsupportedMediaType

Scope: entity

Description: trigger for an HTTP message reception event with status code 415

### c\_EPTF\_HTTP\_inputIdx\_response416RequestedRangeNotSatisfiable

Scope: entity

Description: trigger for an HTTP message reception event with status code 416

### c\_EPTF\_HTTP\_inputIdx\_response417ExpectationFailed

Scope: entity

Description: trigger for an HTTP message reception event with status code 417

### c\_EPTF\_HTTP\_inputIdx\_response500InternalServerError

Scope: entity

Description: trigger for an HTTP message reception event with status code 500

### c\_EPTF\_HTTP\_inputIdx\_response501NotImplemented

Scope: entity

Description: trigger for an HTTP message reception event with status code 501

### c\_EPTF\_HTTP\_inputIdx\_response502BadGateway

Scope: entity

Description: trigger for an HTTP message reception event with status code 502

### c\_EPTF\_HTTP\_inputIdx\_response503ServiceUnavailable

Scope: entity

Description: trigger for an HTTP message reception event with status code 503

### c\_EPTF\_HTTP\_inputIdx\_response504GatewayTimeout

Scope: entity

Description: trigger for an HTTP message reception event with status code 504

### c\_EPTF\_HTTP\_inputIdx\_response505HTTPVersionNotSupported

Scope: entity

Description: trigger for an HTTP message reception event with status code 505

### c\_EPTF\_HTTP\_inputIdx\_response100class

Scope: entity

Description: trigger for an HTTP message reception event with status code in range [100 .. 200)

### c\_EPTF\_HTTP\_inputIdx\_response200class

Scope: entity

Description: trigger for an HTTP message reception event with status code in range [200 .. 300)

### c\_EPTF\_HTTP\_inputIdx\_response300class

Scope: entity

Description: trigger for an HTTP message reception event with status code in range [300 .. 400)

### c\_EPTF\_HTTP\_inputIdx\_response400class

Scope: entity

Description: trigger for an HTTP message reception event with status code in range [400 .. 500)

### c\_EPTF\_HTTP\_inputIdx\_response500class

Scope: entity

Description: trigger for an HTTP message reception event with status code in range [500 .. 600)

### c\_EPTF\_HTTP\_inputIdx\_responseSocketError

Scope: entity

Description: if transport layer cannot open or send message on a socket an IPL4 PortError arrives. This socket error notifies FSM, if any kind of socket error occurred.

## FSMs

No default FSMs provided by the application library.

## Statistics

The application library provides the following statistics:

### c\_EPTF\_HTTP\_Stat\_nofIncRequests

Description: number of incoming requests.

### c\_EPTF\_HTTP\_Stat\_nofIncOptionMessages

Description: number of incoming messages with OPTIONS method

### c\_EPTF\_HTTP\_Stat\_nofOutOptionMessages

Description: number of outgoing messages with OPTIONS method

### c\_EPTF\_HTTP\_Stat\_nofIncGetMessages

Description: number of incoming messages with GET method

### c\_EPTF\_HTTP\_Stat\_nofOutRequests

Description: number of outgoing requests.

### c\_EPTF\_HTTP\_Stat\_nofOutGetMessages

Description: number of outgoing messages with GET method

### c\_EPTF\_HTTP\_Stat\_nofIncHeadMessages

Description: number of incoming messages with HEAD method

### c\_EPTF\_HTTP\_Stat\_nofOutHeadMessages

Description: number of outgoing messages with HEAD method

### c\_EPTF\_HTTP\_Stat\_nofIncPortMessages

Description: number of incoming messages with POST method

### c\_EPTF\_HTTP\_Stat\_nofOutPostMessages

Description: number of outgoing messages with POST method

### c\_EPTF\_HTTP\_Stat\_nofIncPutMessages

Description: number of incoming messages with PUT method

### c\_EPTF\_HTTP\_Stat\_nofOutPutMessages

Description: number of outgoing messages with PUT method

### c\_EPTF\_HTTP\_Stat\_nofIncDeleteMessages

Description: number of incoming messages with DELETE method

### c\_EPTF\_HTTP\_Stat\_nofOutDeleteMessages

Description: number of outgoing messages with DELETE method

### c\_EPTF\_HTTP\_Stat\_nofIncTraceMessages

Description: number of incoming messages with TRACE method

### c\_EPTF\_HTTP\_Stat\_nofOutTraceMessages

Description: number of outgoing messages with TRACE method

### c\_EPTF\_HTTP\_Stat\_nofIncConnectMessages

Description: number of incoming messages with CONNECT method

### c\_EPTF\_HTTP\_Stat\_nofOutConnectMessages

Description: number of outgoing messages with CONNECT method

### c\_EPTF\_HTTP\_Stat\_nofIncResponses

Description: number of incoming responses.

### c\_EPTF\_HTTP\_Stat\_nofIncXXXStatusCode

Description: number of incoming messages with XXX status code where XXX denotes a valid status code (valid status codes are the ones defined in section 3.4)

### c\_EPTF\_HTTP\_Stat\_nofOutResponses

Description: number of outgoing responses.

### c\_EPTF\_HTTP\_Stat\_nofOutXXXStatusCode

Description: number of outgoing messages with XXX status code where XXX denotes a valid status code (valid status codes are the ones defined in section 3.4)