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

ICAP Protocol Module for TTCN-3 Toolset with TITAN, Description

**Abstract**

The purpose of this document is to specify the content of The Internet Content Adaption Protocol (ICAP) protocol module. [1]

Contents

[1 About this Document 2](#_Toc358108449)

[1.1 How to Read this Document 2](#_Toc358108450)

[1.2 Presumed Knowledge 2](#_Toc358108451)

[2 Functionality 2](#_Toc358108452)

[2.1 Protocol version implemented 2](#_Toc358108453)

[2.2 Routing Functionality 2](#_Toc358108454)

[2.3 Modified and non-implemented Protocol Elements 2](#_Toc358108455)

[2.3.1 Relaxed conditions 2](#_Toc358108456)

[2.4 Ericsson-specific changes 2](#_Toc358108457)

[2.5 Backward incompatibilities 2](#_Toc358108458)

[2.6 System Requirements 3](#_Toc358108459)

[3 Feature list 3](#_Toc358108460)

[3.1 Encoding/Decoding and Other Related Functions 3](#_Toc358108461)

[3.1.1 Message length function 3](#_Toc358108462)

[4 Protocol Modules 3](#_Toc358108463)

[4.1 Overview 3](#_Toc358108464)

[4.2 Installation 4](#_Toc358108465)

[4.3 Configuration 4](#_Toc358108466)

[4.4 Module parameters 4](#_Toc358108467)

[4.5 Parser generation rules 4](#_Toc358108468)

[5 Terminology 4](#_Toc358108469)

[5.1 Abbreviations 4](#_Toc358108470)

[6 References 4](#_Toc358108471)

[7 Change information 5](#_Toc358108472)

[7.1 R1A 5](#_Toc358108473)

# About this Document

## How to Read this Document

This is the Description for the Internet Content Adaption Protocol (ICAP) protocol module. The Internet Content Adaption Protocol (ICAP) protocol module is developed for the TTCN-3 Toolset with TITAN. This document should be read together with Product Revision Information [3].

## Presumed Knowledge

To use this protocol module the knowledge of the TTCN-3 language [4] is essential.

# Functionality

The protocol module implements the message structure of the Internet Content Adaption Protocol (ICAP), using the standard specification language TTCNv3. This allows defining of test data in the TTCNv3 language [4] and correctly encoding/decoding these messages when executing test suites using the TITAN TTCNv3 test environment.

## Protocol version implemented

This set of protocol modules implements protocol messages and constants of RFC 3507 [1] and RFC 2616. [2]

## Routing Functionality

Routing functionality is not performed.

## Modified and non-implemented Protocol Elements

### Relaxed conditions

There is no constraint between received and sent messages.

## Ericsson-specific changes

There is no Ericsson specific change in this product.

## Backward incompatibilities

-

## 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 R7A (1.7.pl0) or higher installed. For installation guide see [5]. Please note: This version of the protocol module is not compatible with TITAN releases earlier than R7A.

# Feature list

## Encoding/Decoding and Other Related Functions

This product also contains encoding/decoding functions that assure correct encoding of messages when sent from TITAN and correct decoding of messages when received by TITAN. Implemented encoding/decoding functions:

Name Type of formal parameters Type of return value

**f\_ICAP\_Enc\_binary (in PDU\_ICAP pl\_msg) octetstring;**

**f\_ICAP\_Dec\_binary (in octetstring pl\_stream)**

**PDU\_ICAP;**

**f\_ICAP\_MessageLength(in octetstring pl\_stream)**

**integer;**

### Message length function

The f\_ICAP\_MessageLength function returns the length of the ICAP message from an octetstring. If the length cannot be determined, then it returns the value -1.

# Protocol Modules

## Overview

Protocol modules implement the message 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 [4] and correctly encoding/decoding messages when executing test suites using the Titan TTCN-3 test environment.

## Installation

The set of protocol modules can be used for 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 ICAP protocol is used as a part of a TTCN-3 test suite, this requires Titan 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 [5].

## Configuration

## Module parameters

No module parameters are used in the ICAP protocol module.

## Parser generation rules

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

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

bison -dv -p ICAP \_parse\_ -b ICAP \_parse\_ $<

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

flex -Cfr -8 -Bvpp -P ICAP \_parse\_ ICAP.l

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 used.

## Abbreviations

ETSI European Telecommunications Standards Institute

IETF Internet Engineering Task Force

ICAP Internet Content Adaption Protocol

TTCNv3 Testing and Test Control Notation version 3

# References

1. RFC 3507  
   Internet Content Adaption Protocol
2. RFC 2616  
   Hypertext Transfer Protocol
3. 109 21-CNL 113 779-1 Uen Rev. A  
   The Internet Content Adaption Protocol (ICAP) Protocol Module for TTCN-3 Toolset with TITAN, Product Revision Information
4. ETSI ES 201 873-1 v.3.2.1 (02/2007)  
   The Testing and Test Control Notation version 3. Part 1: Core Language
5. 1/198 17-CRL 113 200/3 Uen  
   User Guide for the TITAN TTCN-3 Test Executor

# Change information

## R1A

Initial version.