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

WebSocket 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 WebSocket Protocol Module.

Contents

[1 Functionality 2](#_Toc356550104)

[1.1 Implemented protocols 2](#_Toc356550105)

[1.2 System Requirements 2](#_Toc356550106)

[1.3 Installation 2](#_Toc356550107)

[1.4 Encoder, decoder functions 2](#_Toc356550108)

[1.5 Masking of the Payload data 3](#_Toc356550109)

[1.6 Masking key generation 3](#_Toc356550110)

[1.7 Message length calculation function 3](#_Toc356550111)

[2 References 3](#_Toc356550112)

# How to Read this Document

This is the Function Description for the WebSocket Protocol Module. The WebSocket Protocol Module is developed for the TTCN-3 Toolset with TITAN. This document should be read together with Product Revision Information [2].

# Functionality

The WebSocket protocol module provides type definitions, encoder, and decoder functions to handle WebSocket messages defined by the standard [3].

## Implemented protocols

The WebSocket protocol module implements all the messages, and information elements defined in RFC 6455 [3].

## System Requirements

In order to operate the WebSocket test port the following system requirements must be satisfied:

* TITAN TTCN-3 Test Executor version R8B (1.8.pl1) or higher installed. Please note: This version of the test port is not compatible with TITAN releases earlier than R8B.

## Installation

Since the WebSocket test port is used as a part of the TTCN-3 test environment this requires TTCN-3 Test Executor to be installed before any operation of the WebSocket test port. For more details on the installation of TTCN-3 Test Executor see the relevant section of [4].

## Encoder, decoder functions

The WebSocket protocol module declares the following encoder, and decoder functions:

external function f\_WebSocket\_Encode(  
in WebSocket\_PDU pl\_pdu, out octetstring pl\_data,  
in boolean pl\_gen\_maks:=m\_Websocket\_generate\_masking\_key, in boolean pl\_auto\_maks:= m\_Websocket\_auto\_masking);

external function f\_WebSocket\_Decode(  
in octetstring pl\_data, out WebSocket\_PDU pl\_pdu, in boolean pl\_auto\_maks:= m\_Websocket\_auto\_masking) return integer;

## Masking of the Payload data

The encoder and decoder functions are able to mask or unmask the Payload data of the WebSocket messages. The functionality is controlled by the pl\_auto\_maks parameter. If the pl\_auto\_maks is true and masking key is present in the message the payload data is masked or unmasked.

## Masking key generation

The masking key can be generated by the following function or auto generated by the f\_WebSocket\_Encode function during encoding.

external function f\_WebSocket\_Generate\_Masking\_Key() return octetstring;

The function returns a randomly generated 4 octet length octestring.

The f\_WebSocket\_Encode function automatically generates and inserts the masking key into the message if:

1. The pl\_gen\_maks parameter is true
2. And mask\_bit == ‘1’B
3. And masking\_key is omit or masking\_key==‘00000000’O

## Message length calculation function

The following function can be used to calculate the length of the received message. The function returns the length of the received message in octets or -1 if the length can’t be calculated.

external function f\_WebSocket\_calc\_length(in octetstring pl\_data) return integer;

# References

1. ETSI ES 201 873-1 v4.5.1   
   The Testing and Test Control Notation version 3. Part 1: Core Language
2. 109 21-CNL 113 782-1  
   WebSocket Protocol Module for TTCN-3 Toolset with TITAN, Product Revision Information
3. RFC 6455  
   The WebSocket Protocol
4. 2/198 17-CRL 113 200/3 Uen  
   Programmer’s Technical Reference for TITAN TTCN–3 Test Executor