Message Session Relay Protocol (MSRP) Protocol Module for TTCN-3 Toolset with TITAN, Description

János Kövesdi

Version 1551-CNL 113 467, Rev. B, 2013-04-17

Table of Contents

How to Read This Document

This is the User Guide for the Message Session Relay Protocol (MSRP) protocol module. The Message Session Relay Protocol (MSRP) protocol module is developed for the TTCN-3 Toolset with TITAN.

Presumed Knowledge

To use this protocol module the knowledge of the TTCN-3 language [1] is essential.

Functionality

The protocol module implements the message structure of the Message Session Relay Protocol (MSRP), using the standard specification language TTCNv3. This allows defining of test data (templates) in the TTCNv3 language [2] and correctly encoding/decoding these messages when executing test suites using the TITAN TTCNv3 test environment.

Protocol module is using TITAN'S TEXT encoding attributes [3] and hence is usable with the TITAN test toolset only.

Protocol Version Implemented

This set of protocol modules implements protocol messages and constants of RFC4975.

Routing Functionality

Routing functionality is not performed.

Modified and Non-Implemented Protocol Elements

Relaxed Conditions

There is no constraint between received and sent messages.

The terminology of v19 is applied for both protocol versions.

Ericsson-Specific Changes

There is no Ericsson specific change in this product.

Backward Incompatibilities

None.

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 [2].

NOTE	This version of the protocol module is not compatible with TITAN releases earlier than R7A.

Feature List

Comparison of Version 12, Version 19 and the Implemented Version

Item/field	V12 specification	V19 specification	Implemented version
`Userinfo`	userid:[pwd], pwd prohibited (":" prohibited) (RFC 3986)	userid:[pwd], pwd prohibited (":" prohibited) (RFC 3986)	V12=v19
`hostport`	host[:port] (RFC3261)	See authority v19	See authority v19
`MSRP-URL vs. MSRP-URI`	MSRP-URL = msrp-scheme "://" [userinfo "@"] hostport ["/" session-id] ";" transport *(";" url-parameter), where transport = "tcp", url-parameter not used in examples	MSRP-URL = msrp-scheme "://" authority ["/" session-id] ";" transport *(";" URI-parameter), where transport ="tcp", url-parameter not used in examples	V19
`url-parameter`	url-parameter = token ["=" token]	Renamed for URI-parameter	Use as URI-parameter
`Authority`	Not specified	authority = [userinfo "@"] host [":" port] (RFC3986)	V19
`URI-parameter`	Not specified	URI-parameter = token ["=" token]	V19

Item/field	V12 specification	V19 specification	Implemented version
namespace	"3 DIGIT"	"3 DIGIT"	Implemented as charstring and restricted for digits at decoding
Session-id	session-id = 1*(unreserved / "+" / "=" / "/")	RFC3986 + "min 80bits" see Ch 14.1 in draftv19, i.e. min 10 bytes	Restricted in receiving templates by version parameters
Transaction-id	Min 4 bytes	min 64 bits according to Ch 7.1 of draftv19 i.e 8 bytes	Restricted in receiving templates by version parameters

IPv6 Address Encoding

The protocol module supports the automatic square bracket adding/removing if needed. The behavior is controlled by module parameter and function parameter.

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_MSRP_Enc	(in PDU_MSRP pl_pdu in in boolean pl_IPv6_bracketing:=tsp_MSRP_EncDec_IPv6_bracketing)	charstring;
f_MSRP_Dec	(in charstring pl_stream) in in boolean pl_IPv6_bracketing:=tsp_MSRP_EncDec_IPv6_bracketing)	PDU_MSRP;
f_MsrpUri_Enc	(in MsrpUri pl_pdu in in boolean pl_IPv6_bracketing:=tsp_MSRP_EncDec_IPv6_bracketing)	charstring;
f_MsrpUri_Dec	(in charstring pl_stream) in in boolean pl_IPv6_bracketing:=tsp_MSRP_EncDec_IPv6_bracketing)	MsrpUri;

Name	Type of formal parameters	Type of return value
f_MSRP_Enc_binary	(in PDU_MSRP pl_msg, in octetstring pl_content_data, in boolean pl_IPv6_bracketing:=tsp_MSRP_EncDec_IPv6_bracketing)	octetstring;
f_MSRP_Dec_binary	(in octetstring pl_stream,out octetstring pl_content_data, in boolean pl_IPv6_bracketing:=tsp_MSRP_EncDec_IPv6_bracketing)	PDU_MSRP;
f_MSRPmsg_MessageLength	(in octetstring pl_stream)	return integer;

The binary encoder and decoder have an additional parameter for handling the binary data in the MSRP message. The binary encoder generates an octetstring from the MSRP PDU and changes the content data part to the received `pl_content_data` binary string. The binary decoder generates an MSRP PDU. It sends the binary content to the `pl_content_data` out parameter and if the `pl_content_data` cannot be used as charstring, then the function changes the content data part of the PDU to an empty string.

The `f_MSRPmsg_MessageLength` function returns the length of the first complete message in the input octetstring. If it does not find an MSRP message in the string, it returns with the value `-1`. The function does not check if the message is correct!

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 [1] and correctly encoding/decoding messages when executing test suites using the Titan TTCN-3 test environment.

Protocol module uses TITAN'S TEXT encoding attributes [2] and hence is usable with the TITAN test toolset only.

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 MSRP 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 [2].

Configuration

Module Parameters

There is a boolean module parameter for debugging purposes of encode/decode functions:

- `tsp_MSRP_EncDec_debug`

  Its default value is `false`, thus in order to have debug information about encoding and decoding it must be set to `true` in the test suite configuration file in the `[MODULE_PARAMETERS]` section.

- `tsp_MSRP_EncDec_IPv6_bracketing`

  It controls the automatic bracketing of IPv6 addresses. Its default value is `true`, thus in order to disable the automatic bracketing it must be set to `false` in the test suite configuration file in the `[MODULE_PARAMETERS]` section.

Parser Generation Rules

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

```
MSRP_parse_.tab.c MSRP _parse_.tab.h: MSRP.y
bison -dv -p MSRP _parse_ -b MSRP _parse_ $<
lex. MSRP _parse_.c: MSRP.l
flex -Cfr -8 -Bvpp -P MSRP _parse_ MSRP.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.

Examples

The "demo" directory of the deliverable contains symlinks for the files of the src directory and additionally *MSRP_Demo.ttcn*, *MSRP.cfg* and *Makefile*.

`MSRP_Demo` Module

The module contains simple examples how to use templates and provides as a basic test of the protocol. It encodes and decodes the MSRP messages, compares the receiving and sending templates with each other and logs the results of the matching.

It contains the following test cases:

```
tc_CheckMsrpSendRequestTemplate_Normal
tc_CheckMsrpResponseTemplate_Normal
tc_CheckMsrpSendRequestTemplate_CpimChunkedData
tc_CheckMsrpSendRequestTemplate_SessionIdLengthTest
tc_CheckMsrpReportRequestTemplate_Normal
tc_CheckMsrpSendRequestTemplate_tooShortMessageId
```

Makefile

It is the *Makefile* that compiles the contents of the demo directory together.

Configuration File

MSRP.cfg contains the config information for the example.

For `tsp_MSRP_EncDec_debug` see Ericsson-Specific Changes.

Module parameter `tsp_requestInfo_Alice2Bob_Normal` provides default information for creating message "MSRP request".

```
[MODULE_PARAMETERS]
tsp_MSRP_EncDec_debug := true,
tsp_requestInfo_Alice2Bob_SEND := {…}
```

For explanation other config file parameters see [3].

How to Use Template Generating Functions

The demo module provides simple examples of how to use template variable generating functions and version handling. The basic idea is to pass all field values different from the default value for the generating functions in a single structure (`Struct_MsrpRequestInfo` and `Struct_MsrpResponseInfo`). The fields of these structures have the default value `"omit"`. They are initialized by dedicated functions. A field having the value `"omit"` orders the generating function not to omit this field but to use the default value of this field. If the user wants to modify this default value, he modifies the field of the structure for the desired value. After the template generating function has exited, the `"return"` value can be modified. This way the user has maximal flexibility with a minimum of effort.

The other parameter of the template generating functions contains the version information. For MSRP it is especially useful for receiving templates because the requirements can be ordered according to different specifications.

The module also demonstrates how module parameters can be used for supporting template creation.

Terminology

No specific terminology used.

Abbreviations

ETSI
: European Telecommunications Standards Institute

IETF
: Internet Engineering Task Force

MSRP
: Message Session Relay Protocol

TTCNv3
: Testing and Test Control Notation version 3

Change Information

R7B

Changed the description of the binary decoder.

References

[1] ETSI ES 201 873-1 v.3.2.1 (02/2007)
The Testing and Test Control Notation version 3. Part 1: Core Language

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

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

[4] RFC4975
The Message Session Relay Protocol (MSRP)