H.248 V2 Protocol Modules for TTCN-3 Toolset with TITAN, User Guide

Gábor Szalai

Version 198 17-CNL 113 424, Rev. H, 2014-07-07

Table of Contents

About This Document

How to Read This Document

This is the User Guide for the H.248 protocol module. The H.248 protocol module is developed for the TTCN-3 Toolset with TITAN. This document should be read together with Function Specification [5].

Presumed Knowledge

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

Basic knowledge of the H.248 V2 and V3 protocol [6], [7] and [9] and SDP [8] is valuable to use this protocol module.

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 R8B (1.8.pl1) or higher installed. For installation guide see [4]. Please note: This version of the protocol module is not compatible with TITAN releases earlier than R8B

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 [3] for SDP encoding and decoding 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 H.248 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 [4].

In order to build an executable with H248 protocol module the following files need to be added to the project file (see 4.3 of [2]) or the *Makefile* (see 12.1.1 of [2]):

- *H248_Types.ttcn*
- *H248_SDP_Types.ttcn*
- *H248_EncDec.cc*
- *H248_SDP_EncDec.cc*
- *H248_p_types.hh*
- *H248_p.hh*
- *H248_p.cc*
- *H248_la.cc*

Configuration

The protocol module has the following module parameters:

- `par_H248_EncDec_debug`

  These Boolean module parameter controls the debug functionality of the decoder. Its default value is `false`, thus in order to have debug information it must be set to `true` in the test suite configuration file in the `[MODULE_PARAMETERS]` section.

- `par_H248_Enc_header_format`

  These enumerated module parameter controls the encoded format of the H248 tokens.

  Possible values:

  - `H248_LONG_TOKENS` - The encoder use long token names. Default.
  - `H248_SHORT_TOKENS` - The encoder use the short format of the tokens.
  - `H248_RANDOM_TOKENS` - The encoder use both sort and long formats of the token in random way.

```
[MODULE_PARAMETERS]
par_H248_EncDec_debug := true,
par_H248_Enc_header_format := H248_SHORT_TOKENS
...
```

Upgrading from previous version

The type structure has changed between R1A04 and R1A05 that causes backward incompatibilities

in the TTCN-3 type definition module. With the updated protocol module both V2 and V3 version of H248 messages can be used with the limitation listed in clause 3.2 of [5].

When upgrading from version R1A04 the following files need to be added to the project file (see 4.3 of [2]) or the *Makefile* (see 12.1.1 of [2]) of the existing files (theses files are needed for decoding using flex/bison parser):

- *H248_p_types.hh*
- *H248_p.hh*
- *H248_p.cc*
- *H248_la.cc*

Functions and templates using the types of H248 module R1A04 need to be updated according to the changes of the type definition.

In case if new fields were added into existing record or set types, the new templates should contain these fields set to omit.

In case if a type has changed completely the whole template or part of template must be changed.

If a function is accessing a field that has changed that function needs to be updated as well.

Here you can find a list of changes within the type definition module of H248 from R1A04 to R1A05.

1. A new parameter called `par_H248_Enc_header_format` is introduced.
2. All text coding attributes have been removed.
3. `SegmentReply` is a new type in the module. It was added as a new field called `segmentReply` to the Transaction union type.
4. Mistyped field `greathethan` was corrected to `greaterthan` in the type `ParmValue`. Templates or function referred to this name need to be corrected as well.
5. In the set type `ContextRequest` two new optional fields have been added. These fields are `iepsValue` and `contextAttrDescriptor`. To update a template containing this type, the new fields have to be added and shall be set to omit.
6. Two new enumerated values have been added to the `TopologyDirection` type. These are: `onewayExternal` and `onewayBoth`.
7. The set type `ContextAttrAuditRequest` has been changed. The type of existing fields `topology`, `emergency` and `priority` have been changed from boolean to `H248_token` type. In templates you should update the value of this field to `present` instead of `true` and `omit` instead of `false`. The following new fields have been added to the set: `priorityValue`, `emergencyValue`, `iepsValue`, `contextAttrDescriptor`, `auditSelectionLogic`, `iEPS` and `pkgdName`. To update a template containing this type, the new fields have to be added and shall be set to omit.
8. A new alternative `statisticsDiscriptor` was added to the union type `AmmDescriptor`.
9. Two new optional fields `–segmentNumber` and `segmentationComplete-` were added to the `TransactionReply` type. To update a template containing this type, the new fields have to be added and shall be set to omit.

10. The mandatory `contextBody` field of `ActionReply` has been changed to optional.
11. The type named `AuditToken` was renamed to `AuditTokens`. The type name shall be updated when referred.
12. The type `IndAudTerminationStateDescriptor` has completely changed. Templates using this type need to be updated accordingly.
13. The `multiStream` alternative of Stream type now contains a record of `IndAudStreamDescriptor` type instead of a single value. In order to update the templates based on this type, an extra {..} need to be added around a single `IndAudStreamDescriptor` record.
14. The following new optional fields have been added to the `IndAudStreamParms` type: `localDescriptor`, `remoteDescriptor` and `statisticsDescriptor`. To update a template contain this type, the new fields have to be added and shall be set to `omit`.
15. The type `IndAudLocalControlDescriptor` has completely changed. Templates using this type need to be updated accordingly.
16. The type `IndAudSignalsDescriptor` has completely changed. Templates using this type need to be updated accordingly.
17. In the `RequestedEvent` type the `eventParameters` field has changed from a `record of union` type to a `set` type. As a result the outer {...} shall be removed from the templates, and all elements of the set must be listed and set to omit if not present. Additionally, some new elements were added to the set, according to the version 3 of H.248.
18. Similar changes listed in item 17 have been done on type `SecondRequestedEvent`.
19. In the set type `ServiceChangeDescriptor` a new optional field `serviceChangeIncomplete` has been added. To update a template contain this type, the new fields has to be added and shall be set to omit.
20. The set type `SignalParams` has been changed. The type of existing field `priority` has been changed from boolean to `H248_token` type. In templates you should update the value of this field to `present` instead of `true` and `omit` instead of `false`. The following new fields have been added to the set: `sigDirection` and `sigRequestId`. To update a template containing this type, the new fields have to be added and shall be set to `omit`.
21. The set type `NotifyCompletion` has been changed. The type of all existing fields have been changed from boolean to `H248_token` type. In templates you should update the value of these fields to `present` instead of `true` and `omit` instead of `false`. A new field called `onIteration` has been added to the set. To update a template containing this type, the new field has to be added and shall be set to omit.
22. In the record type `StatisticsParameter` the field `statValue` has been renamed to `values` and its type has been changed to `record of Value` instead of a single `Value` type.
23. The type of `streamParms` field of `StreamDescriptor` type has been changed to a record of `StreamParm` instead of a single `StreamParm`.
24. A new optional `statiscticsDescriptor` field to the `StreamParm` set type. To update a template containing this type, the new field has to be added and shall be set to omit.

Examples

The "demo" directory of the deliverable contains the following examples and functions:

Mapping Module

The mapping module provides the connection between the H.248 protocol module and the TCP test port (CNL 113 347). It encodes and decodes the H.248 messages.

SDP Parsing

Use the `f_H248_SDP_Dec` function to decode a charstring SDP message list from the Local/Remote descriptor value.

For encoding use the `f_H248_SDP_Enc` function to encode a SDP message list to charstring, and put it into the `Local`/`Remote` descriptor field.

```
var H248_SDP_Message_list v_H248_SDP_Messages;
var charstring v_descriptor;
// put encoded value in v_descriptor
v_H248_SDP_Messages := f_H248_SDP_Dec(v_descriptor);
v_descriptor := f_H248_SDP_Enc(v_H248_SDP_Messages);
```

Parser Generation Rules

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

```
H248_SDP_parse_.tab.c H248_SDP_parse_.tab.h: H248_SDP_parser.y
bison -dv -p H248_SDP_parse_ -b H248_SDP_parse_ $<
lex.H248_SDP_parse_.c: H248_SDP_parser.l
flex -Cfr -8 -Bvpp -PH248_SDP_parse_ H248_SDP_parser.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 is used.

Abbreviations

TTCN-3

Testing and Test Control Notation version 3

References

[1] ETSI ES 201 873-1 v.2.2.1 (02/2003)
The Testing and Test Control Notation version 3. Part 1: Core Language

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

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

[4] Installation Guide for the TITAN TTCN-3 Test Executor

[5] H.248 V2 Protocol Modules for TTCN-3 Toolset with TITAN, Function Specification

[6] H.248.1 (05/2002)
Gateway control protocol: Version 2

[7] H.248.1 v2 Corrigendum 1 (03/2004)
Gateway control protocol: Version 2 Corrigendum 1

[8] RFC 2327
SDP: Session Description Protocol

[9] H248.1 (08/2005)
Gateway control protocol: Version 3 Draft