****

**ZigBee Cluster Library  
Thermostat Cluster (0x0201)   
Test Specification  
Version 0.9**

|  |  |
| --- | --- |
| ZigBee Document 16-02865-001 | |
| October 5th, 2017 | |
| Sponsored by: ZigBee Alliance | |
| Accepted by | This document has not yet been accepted for release by the ZigBee Alliance Board of Directors |
| Abstract | This document describes the certification tests for devices which implement the ZCL Thermostat cluster. |
| Keywords | ZCL, Thermostat, cluster |

This page is intentionally blank

Notice of use and disclosure

Copyright © ZigBee Alliance, Inc. (1996-2019). All rights Reserved. This information within this document is the property of the ZigBee Alliance and its use and disclosure are restricted.

Elements of ZigBee Alliance specifications may be subject to third party intellectual property rights, including without limitation, patent, copyright or trademark rights (such a third party may or may not be a member of ZigBee). ZigBee is not responsible and shall not be held responsible in any manner for identifying or failing to identify any or all such third party intellectual property rights.

No right to use any ZigBee name, logo or trademark is conferred herein.  Use of any ZigBee name, logo or trademark requires membership in the ZigBee Alliance and compliance with the ZigBee Logo and Trademark Policy and related ZigBee policies.

This document and the information contained herein are provided on an “AS IS” basis and ZigBee DISCLAIMS ALL WARRANTIES EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO (A) ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OF THIRD PARTIES (INCLUDING WITHOUT LIMITATION ANY INTELLECTUAL PROPERTY RIGHTS INCLUDING PATENT, COPYRIGHT OR TRADEMARK RIGHTS) OR (B) ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE OR NONINFRINGEMENT. IN NO EVENT WILL ZIGBEE BE LIABLE FOR ANY LOSS OF PROFITS, LOSS OF BUSINESS, LOSS OF USE OF DATA, INTERRUPTION OF BUSINESS, OR FOR ANY OTHER DIRECT, INDIRECT, SPECIAL OR EXEMPLARY, INCIDENTIAL, PUNITIVE OR CONSEQUENTIAL DAMAGES OF ANY KIND, IN CONTRACT OR IN TORT, IN CONNECTION WITH THIS DOCUMENT OR THE INFORMATION CONTAINED HEREIN, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE. All Company, brand and product names may be trademarks that are the sole property of their respective owners.

The above notice and this paragraph must be included on all copies of this document that are made.

This page is intentionally blank

Revision history

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Date | Details | Editor |
| 000 | September 21st, 2016 | First draft. | Bozena Erdmann |
| 001 | October 5th, 2017 | Fixed CCB #2461. | Phil Jamieson |

This page is intentionally blank

Table of Contents

[1 Introduction 9](#_Toc494983535)

[1.1 Conformance levels 9](#_Toc494983536)

[2 References 10](#_Toc494983537)

[2.1 ZigBee Alliance documents 10](#_Toc494983538)

[2.2 IETF documents 10](#_Toc494983539)

[3 PICS 11](#_Toc494983540)

[3.1 Usage 11](#_Toc494983541)

[3.2 Server 11](#_Toc494983542)

[3.2.1 Attributes 11](#_Toc494983543)

[3.2.2 Commands received 14](#_Toc494983544)

[3.2.3 Commands sent 15](#_Toc494983545)

[3.3 Client 15](#_Toc494983546)

[3.3.1 Attributes 15](#_Toc494983547)

[4 Test specification 16](#_Toc494983548)

[4.1 Introduction 16](#_Toc494983549)

[4.1.1 Test case overview 16](#_Toc494983550)

[4.1.2 Testing tolerances 16](#_Toc494983551)

[4.1.3 Client DUTs 16](#_Toc494983552)

[4.1.4 Test steps manipulating attributes 17](#_Toc494983553)

[4.2 Generic test cases 18](#_Toc494983554)

[4.2.1 TSTAT-TC-01G: Global attributes 18](#_Toc494983555)

[4.3 Server test cases 22](#_Toc494983556)

[4.3.1 TSTAT-TC-01S: Attributes with server as DUT 22](#_Toc494983557)

[4.3.2 TSTAT-TC-02S: Setpoint Test Cases with server as a DUT 27](#_Toc494983558)

[4.3.3 TSTAT-TC-03S: Schedule test cases with server as a DUT 56](#_Toc494983559)

[4.3.4 TSTAT-TC-04S: Thermostat cluster with separate temperature sensor and HVAC unit functionality with server as DUT 64](#_Toc494983560)

[4.3.5 TSTAT-TC-05S: Scenes functionality with server as DUT (!!!) 71](#_Toc494983561)

[4.3.6 TSTAT-TC-06S: Reporting functionality with server as DUT 75](#_Toc494983562)

[4.4 Client test cases 81](#_Toc494983563)

[4.4.1 TSTAT-TC-01C: Functionality with client as DUT (TBD) 81](#_Toc494983564)

[5 Annex A: PICS to test case cross reference (TBD) 84](#_Toc494983565)

[5.1 Server 84](#_Toc494983566)

[5.2 Client 85](#_Toc494983567)

This page is intentionally blank

# Introduction

This document contains the PICS, test specification and PICS/test case cross reference for the ZCL *thermostat* cluster.

## Conformance levels

The key words "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED" and "MAY" in this document are to be interpreted as described in [R4].

# References

## ZigBee Alliance documents

1. ZigBee Cluster Library Specification, ZigBee Alliance document 07-5123.
2. ZCL General Test Specification, ZigBee Alliance document 16-0xxx.
3. ZCL Thermostat Cluster XML PICS, ZigBee Alliance document 16-0xxx.

## IETF documents

1. S. Bradner, Key words for use in RFCs to Indicate Requirement Levels, IETF RFC 2119, March 1997.

# PICS

All references are for the ZigBee Cluster Library specification [R1] unless otherwise indicated. An XML version of these PICS is also available in [R3].

## Usage

| Item number | Feature | Reference | Status | Support |
| --- | --- | --- | --- | --- |
| TSTAT.S | Does the device implement the *thermostat* cluster as a server? | 3.3.2 | O | Yes |
| TSTAT.C | Does the device implement the *thermostat* cluster as a client? | 3.3.3 | O | No |

## Server

### Attributes

| Item number | Feature | Reference | Status | Support |
| --- | --- | --- | --- | --- |
| Thermostat Information Attribute Set | | | | |
| TSTAT.S.A0000 | Does the device implement the *LocalTemperature* attribute? | Table 6.11, 6.3.2.2.1.1 | TSTAT.S: M | Yes |
| TSTAT.S.A0000.Repor t.DefaultConfig | Does the device implement default reporting configuration for the *LocalTemperature* attribute? | 6.3.2.5 | TSTAT.S: M | Yes |
| TSTAT.S.A0000.Report.Tx | Does the device implement receiving and responding to the global report attribute commands for the *LocalTemperature* attribute and sending reports? | 6.3.2.5 | TSTAT.S: M | Yes |
| TSTAT.S.A0001 | Does the device implement the *OutdoorTemperature* attribute? | Table 6.11, 6.3.2.2.1.2 | TSTAT.S: O | No |
| TSTAT.S.A0002 | Does the device implement the *Occupancy* attribute? | Table 6.11, 6.3.2.2.1.3 | TSTAT.S: O | No |
| TSTAT.S.A0003 | Does the device implement the *AbsMinHeatSetpointLimit* attribute? | Table 6.11, 6.3.2.2.1.4 | TSTAT.S: O | Yes |
| TSTAT.S.A0004 | Does the device implement the *AbsMaxHeatSetpointLimit* attribute? | Table 6.11, 6.3.2.2.1.5 | TSTAT.S: O | Yes |
| TSTAT.S.A0005 | Does the device implement the *AbsMinCoolSetpointLimit* attribute? | Table 6.11, 6.3.2.2.1.6 | TSTAT.S: O | No |
| TSTAT.S.A0006 | Does the device implement the *AbsMaxCoolSetpointLimit* attribute? | Table 6.11, 6.3.2.2.1.7 | TSTAT.S: O | No |
| TSTAT.S.A0007 | Does the device implement the *PICoolingDemand* attribute? | Table 6.11, 6.3.2.2.1.8 | TSTAT.S: O | No |
| TSTAT.S.A0007.Repor t.DefaultConfig | Does the device implement default reporting configuration for the *PICoolingDemand* attribute? | 6.3.2.5 | TSTAT.S.A0007: M | No |
| TSTAT.S.A0007.Report.Tx | Does the device implement receiving and responding to the global report attribute commands for the *PICoolingDemand* attribute and sending reports? | 6.3.2.5 | TSTAT.S.A0007: M | No |
| TSTAT.S.A0008 | Does the device implement the *PIHeatingDemand* attribute? | Table 6.11, 6.3.2.2.1.9 | TSTAT.S: O | Yes |
| TSTAT.S.A0008.Repor t.DefaultConfig | Does the device implement default reporting configuration for the *PIHeatingDemand* attribute? | 6.3.2.5 | TSTAT.S.A0008: M | Yes |
| TSTAT.S.A0008.Report.Tx | Does the device implement receiving and responding to the global report attribute commands for the *PIHeatingDemand* attribute and sending reports? | 6.3.2.5 | TSTAT.S.A0008: M | Yes |
| TSTAT.S.A0009 | Does the device implement the *HVACSystemTypeConfiguration* attribute? | Table 6.11, 6.3.2.2.1.10 | TSTAT.S: O | No |
| Thermostat Settings Attribute Set | | | | |
| TSTAT.S.A0010 | Does the device implement the *LocalTemperatureCalibration* attribute? | Table 6.13, 6.3.2.2.2.1 | TSTAT.S: O | No |
| TSTAT.S.A0011 | Does the device implement the *OccupiedCoolingSetpoint* attribute? | Table 6.13, 6.3.2.2.2.2 | TSTAT.S: M.1[[1]](#footnote-2) | No |
| TSTAT.S.A0011.Scene | Does the device implement receiving and responding to the scene cluster commands for the *OccupiedCoolingSetpoint* attribute? | 6.3.2.6 | (TSTAT.S.A0011 & S.S): M | No |
| TSTAT.S.A0012 | Does the device implement the *OccupiedHeatingSetpoint* attribute? | Table 6.13, 6.3.2.2.2.3 | TSTAT.S: M.1 | Yes |
| TSTAT.S.A0012.Scene | Does the device implement receiving and responding to the scene cluster commands for the *OccupiedHeatingSetpoint* attribute? | 6.3.2.6 | (TSTAT.S.A0012 & S.S): M | No |
| TSTAT.S.A0013 | Does the device implement the *UnoccupiedCoolingSetpoint* attribute? | Table 6.13, 6.3.2.2.2.4 | TSTAT.S: O | No |
| TSTAT.S.A0014 | Does the device implement the *UnoccupiedHeatingSetpoint* attribute? | Table 6.13, 6.3.2.2.2.5 | TSTAT.S: O | No |
| TSTAT.S.A0015 | Does the device implement the *MinHeatSetpointLimit* attribute? | Table 6.13, 6.3.2.2.2.6 | TSTAT.S: O | Yes |
| TSTAT.S.A0016 | Does the device implement the *MaxHeatSetpointLimit* attribute? | Table 6.13, 6.3.2.2.2.7 | TSTAT.S: O | Yes |
| TSTAT.S.A0017 | Does the device implement the *MinCoolSetpointLimit* attribute? | Table 6.13, 6.3.2.2.2.8 | TSTAT.S: O | No |
| TSTAT.S.A0018 | Does the device implement the *MaxCoolSetpointLimit* attribute? | Table 6.13, 6.3.2.2.2.9 | TSTAT.S: O | No |
| TSTAT.S.A0019 | Does the device implement the *MinSetpointDeadBand* attribute? | Table 6.13, 6.3.2.2.2.10 | TSTAT.S: O | No |
| TSTAT.S.A001A | Does the device implement the *RemoteSensing* attribute? | Table 6.13, 6.3.2.2.2.11 | TSTAT.S: O | No |
| TSTAT.S.A001B | Does the device implement the *ControlSequenceOfOperation* attribute? | Table 6.13, 6.3.2.2.2.12 | TSTAT.S: M | Yes |
| TSTAT.S.A001C | Does the device implement the *SystemMode* attribute? | Table 6.13, 6.3.2.2.2.13 | TSTAT.S: M | Yes |
| TSTAT.S.A001C.Scene | Does the device implement receiving and responding to the scene cluster commands for the *SystemMode* attribute? | 6.3.2.6 | (TSTAT.S.A001C & S.S): M | No |
| TSTAT.S.A001D | Does the device implement the *AlarmMask* attribute? | Table 6.13, 6.3.2.2.2.14 | TSTAT.S: O | No |
| TSTAT.S.A001E | Does the device implement the *ThermostatRunningMode* attribute? | Table 6.13, 6.3.2.2.2.15 | TSTAT.S: O | No |
| Thermostat Schedule & HVAC Relay Attribute Set | | | | |
| TSTAT.S.A0020 | Does the device implement the *StartOfWeek* attribute? | Table 6.20, 6.3.2.2.3.1 | TSTAT.S: O | No |
| TSTAT.S.A0021 | Does the device implement the *NumberOfWeeklyTransitions* attribute? | Table 6.20, 6.3.2.2.3.2 | TSTAT.S: O | No |
| TSTAT.S.A0022 | Does the device implement the *NumberOfDailyTransitions* attribute? | Table 6.20, 6.3.2.2.3.3 | TSTAT.S: O | No |
| TSTAT.S.A0023 | Does the device implement the *TemperatureSetpointHold* attribute? | Table 6.20, 6.3.2.2.3.4 | TSTAT.S: O | No |
| TSTAT.S.A0024 | Does the device implement the *TemperatureSetpointHoldDuration* attribute? | Table 6.20, 6.3.2.2.3.5 | TSTAT.S: O | No |
| TSTAT.S.A0025 | Does the device implement the *ThermostatProgrammingOperationMode* attribute? | Table 6.20, 6.3.2.2.3.6 | TSTAT.S: O | No |
| TSTAT.S.A0029 | Does the device implement the *ThermostatRunningState* attribute? | Table 6.20, 6.3.2.2.3.7 | TSTAT.S: O | No |
| Thermostat Setpoint Change Tracking Attribute Set | | | | |
| TSTAT.S.A0030 | Does the device implement the *SetpointChangeSource* attribute? | Table 6.25, 6.3.2.2.4.1 | TSTAT.S: O | No |
| TSTAT.S.A0031 | Does the device implement the *SetpointChangeAmount* attribute? | Table 6.25, 6.3.2.2.4.2 | TSTAT.S: O | No |
| TSTAT.S.A0032 | Does the device implement the *SetpointChangeSourceTimestamp* attribute? | Table 6.25, 6.3.2.2.4.3 | TSTAT.S: O | No |
| TSTAT.S.A0034 | Does the device implement the *OccupiedSetback* attribute? | Table 6.25, 6.3.2.2.4.4 | TSTAT.S: O | No |
| TSTAT.S.A0035 | Does the device implement the *OccupiedSetbackMin* attribute? | Table 6.25, 6.3.2.2.4.5 | TSTAT.S: O | No |
| TSTAT.S.A0036 | Does the device implement the *OccupiedSetbackMax* attribute? | Table 6.25, 6.3.2.2.4.6 | TSTAT.S: O | No |
| TSTAT.S.A0037 | Does the device implement the *UnoccupiedSetback* attribute? | Table 6.25, 6.3.2.2.4.7 | TSTAT.S: O | No |
| TSTAT.S.A0038 | Does the device implement the *UnoccupiedSetbackMin* attribute? | Table 6.25, 6.3.2.2.4.8 | TSTAT.S: O | No |
| TSTAT.S.A0039 | Does the device implement the *UnoccupiedSetbackMax* attribute? | Table 6.25, 6.3.2.2.4.9 | TSTAT.S: O | No |
| TSTAT.S.A003a | Does the device implement the *EmergencyHeatDelta* attribute? | Table 6.25, 6.3.2.2.4.10 | TSTAT.S: O | No |
| AC Information Attribute Set | | | | |
| TSTAT.S.A0040 | Does the device implement the *ACType* attribute? | Table 6.28, 6.3.2.2.5.1 | TSTAT.S: O | No |
| TSTAT.S.A0041 | Does the device implement the *ACCapacity* attribute? | Table 6.28, 6.3.2.2.5.2 | TSTAT.S: O | No |
| TSTAT.S.A0042 | Does the device implement the *ACRefrigerantType* attribute? | Table 6.28, 6.3.2.2.5.3 | TSTAT.S: O | No |
| TSTAT.S.A0043 | Does the device implement the *ACCompressorType* attribute? | Table 6.28, 6.3.2.2.5.4 | TSTAT.S: O | No |
| TSTAT.S.A0044 | Does the device implement the *ACErrorCode* attribute? | Table 6.28, 6.3.2.2.5.5 | TSTAT.S: O | No |
| TSTAT.S.A0045 | Does the device implement the *ACLouverPosition* attribute? | Table 6.28, 6.3.2.2.5.6 | TSTAT.S: O | No |
| TSTAT.S.A0046 | Does the device implement the *ACCoilTemperature* attribute? | Table 6.28, 6.3.2.2.5.7 | TSTAT.S: O | No |
| TSTAT.S.A0047 | Does the device implement the *ACCapacityFormat* attribute? | Table 6.28, 6.3.2.2.5.8 | TSTAT.S: O | No |
| Global attributes | | | | |
| TSTAT.S.Afffd | Does the device implement the *ClusterRevision* global attribute? | Table 2-1, 2.3.5.1.1 | TSTAT.S: M | Yes |
| TSTAT.S.Afffe | Does the device implement the *AttributeReportingStatus* global attribute? | Table 2-1, 2.3.5.1.2 | TSTAT.S: O | No |

### Commands received

| Item number | Feature | Reference | Status | Support |
| --- | --- | --- | --- | --- |
| TSTAT.S.C00.Rsp | Does the device implement receiving the *Setpoint Raise/Lower* command? | Table 6.35, 6.3.2.3.1 | TSTAT.S: M | Yes |
| TSTAT.S.C01.Rsp | Does the device implement receiving the *Set Weekly Schedule* command? | Table 6.35, 6.3.2.3.2 | TSTAT.S: O | No |
| TSTAT.S.C02.Rsp | Does the device implement receiving the *Get Weekly Schedule* command? | Table 6.35, 6.3.2.3.3 | TSTAT.S: O | No |
| TSTAT.S.C03.Rsp | Does the device implement receiving the *Clear Weekly Schedule* command? | Table 6.35, 6.3.2.3.4 | TSTAT.S: O | No |
| TSTAT.S.C04.Rsp | Does the device implement receiving the *Get Relay Status Log* command? | Table 6.35, 6.3.2.3.5 | TSTAT.S: O | No |

#### Commands sent

| Item number | Feature | Reference | Status | Support |
| --- | --- | --- | --- | --- |
| TSTAT.S.C00.Tx | Does the device implement sending the *Get Weekly Schedule Response* command? | Table 6.39, 6.3.2.4.1 | TSTAT.S.C02.Rsp: M | No |
| TSTAT.S.C01.Tx | Does the device implement receiving the *Get Relay Status Log Response* command? | Table 6.39, 6.3.2.4.2 | TSTAT.S.C04.Rsp: M | No |

## Client

### Attributes

| Item number | Feature | Reference | Status | Support |
| --- | --- | --- | --- | --- |
| TSTAT.C.A0000.Report.Rsp | Does the device implement sending global report attribute command requests and receiving reports for the *MeasuredValue* attribute? | 4.7.2.5 | TSTAT.C: O | No |
| TSTAT.C.A0007.Report.Rsp | Does the device implement sending global report attribute command requests and receiving reports for the *PICoolingDemand* attribute? | 4.7.2.5 | TSTAT.C: O | No |
| TSTAT.C.A0008.Report.Rsp | Does the device implement sending global report attribute command requests and receiving reports for the *PIHeatingDemand* attribute? | 4.7.2.5 | TSTAT.C: O | No |
| TSTAT.C.Afffd | Does the device implement the *ClusterRevision* global attribute? | Table 2-1, 2.3.5.1.1 | TSTAT.C: M | No |
| TSTAT.C.Afffe | Does the device implement the *AttributeReportingStatus* global attribute? | Table 2-1, 2.3.5.1.2 | TSTAT.C: O | No |

# Test specification

## Introduction

### Test case overview

The following test cases are available for the *thermostat* cluster:

|  |  |  |
| --- | --- | --- |
| **Test ID** | **Description** | **Reference** |
| **Global tests** | | |
| TSTAT-TC-01G | Global attributes | 4.2.1 |
| **Server side tests** | | |
| TSTAT-TC-01S | Attributes with server as DUT | 4.3.1 |
| TSTAT-TC-02S | Setpoint test cases with server as DUT | 4.3.2 |
| TSTAT-TC-03S | Schedule test cases with server as DUT | 4.3.3 |
| TSTAT-TC-04S | Thermostat cluster with separate temperature sensor and HVAC unit functionality with server as DUT | 4.3.4 |
| TSTAT-TC-05S | Scenes functionality with server as DUT | 4.3.5 |
| TSTAT-TC-06S | Reporting functionality with server as DUT | 4.3.6 |
| **Client side tests** | | |
| TSTAT-TC-01C | Functionality with client as DUT | 4.4.1 |

### Testing tolerances

In test cases where a change in an attribute value is tested over time, it is permitted for the devices involved in the test to be within a tolerance of ±15% of the expected value. As such, these test cases indicate that the attribute value must be approximately equal to an expected value, to which the ±15% tolerance should then be applied.  All other attribute values presented are expected to be exact.

### Client DUTs

For client test cases only test steps that pertain to commands that are supported on the DUT are required to be executed. All commands in this cluster for which support is indicated in the PICS shall be exercised, using valid, application achievable values.

Note that for the client attribute test case, it is permissible for the client not to be able to execute any of the test steps.

The client SHALL ensure that an application link, e.g. a binding link, exists between itself and the test harness. This should be configured before starting the test.

### Test steps manipulating attributes

In test case steps that require more than one attribute to be manipulated (e.g. read), the tester may decide whether it is appropriate or practical to send a single attribute manipulation command, containing multiple attributes, or multiple attribute manipulation commands, each containing a single attribute. The test case is designed to verify the behavior of the device supporting the attribute rather that verifying the attribute manipulation command in question.

## Generic test cases

### TSTAT-TC-01G: Global attributes

This test case verifies the behavior of the global attributes of the *thermostat* cluster client and server.

In this test, the PICS notation TSTAT.S.A*gm* and TSTAT.C.A*gm* represents the list of global attributes that are specified as being mandatory for either the server or client, respectively. Similarly, the PICS notation TSTAT.S.A*go* and TSTAT.C.A*go* represents the list of global attributes that are specified as being optional for either the server or client, respectively.

#### Scope

General:

* *Read attributes* command (0x00)
* *Read attributes response* command (0x01)
* *Write attributes* command (0x02)
* *Write attributes response* command (0x04)

*Thermostat* cluster (0x0201):

* Allglobal attributes

PICS:

* TSTAT.S, TSTAT.C
* TSTAT.S.A*gm*, TSTAT.C.A*gm*, TSTAT.S.A*go*, TSTAT.C.A*go*

#### Required devices

|  |  |  |
| --- | --- | --- |
| **Designation** | **Symbol** | **Description** |
| DUT |  | Device under test implementing:   * The *thermostat* cluster server or client. |
| TH |  | Test harness implementing:   * The *thermostat* cluster client or server, i.e. the opposite cluster instantiation as implemented on the DUT. |

#### Initial conditions

|  |  |
| --- | --- |
| **Item** | **Initial Conditions** |
| 1 | A packet sniffer shall be observing the communication over the air interface. |
| 2 | All devices are factory new and powered off until used. |

#### Test preparation

|  |  |
| --- | --- |
| **Add TH and DUT to a ZigBee network** | |
|  |  |
| *Before* | *After* |

| **TSTAT-TC-01G: Global attributes** | | |
| --- | --- | --- |
| **Item** | **Preparation Step** | **Observation** |
| P1 | Form a ZigBee network. | Observe appropriate command frame to form the network. |
| P2 | Power on TH and DUT. | TH and DUT are powered on. |
| P3 | Join TH and DUT to a ZigBee network. | Observe appropriate communication between TH, DUT and any other relevant node on the ZigBee network. |
| *--- End of test case TSTAT-TC-01G preparation ---* | | |

#### Test procedure

| **TSTAT-TC-01G: Global attributes** | | | | |
| --- | --- | --- | --- | --- |
| **Item** | **PICS** | **Test Harness Step** | **DUT pass Verification** | |
| 1 | TSTAT.S.A*gm*, TSTAT.C.A*gm* | TH unicasts a ZCL *read attributes* command frame to DUT to read each mandatory global attribute of this cluster one at a time. | DUT unicasts a ZCL *read attributes response* command frame to TH containing each requested attribute.  The data type in each command must match the value listed in the specification(s). The data value in each command for the attribute must fall within the valid range described in the specification(s). | |
| 2a | TSTAT.S.A*gm*, TSTAT.C.A*gm* | TH unicasts a ZCL *write attributes* command frame to DUT to write the respective default value to each mandatory global attribute of this cluster one at a time. | DUT unicasts a ZCL *write attributes response* command frame to TH for each attribute.  If the access control of DUT is set to READ, the DUT response will indicate that the attribute write command was not a SUCCESS. If the access control of DUT is set to READ/WRITE, the DUT response will indicate that the write command was a SUCCESS. | |
| 2b | TSTAT.S.A*gm*, TSTAT.C.A*gm* | TH unicasts a ZCL *read attributes* command frame to DUT to read back each attribute written in step 2a. | DUT unicasts a ZCL *read attributes response* command frame to TH containing the requested attribute.  If the *Status* field of the *write attributes response* command frame was equal to SUCCESS, the updated value is read back. If the *Status* field of the *write attributes response* command frame was not equal to SUCCESS the value is not updated when read back. | |
| *Continued…* | | | | |
| 3 | TSTAT.S.A*go*, TSTAT.C.A*go* | TH unicasts a ZCL *read attributes* command frame to DUT to read each optional global attribute of this cluster one at a time. | DUT unicasts a ZCL *read attributes response* command frame to TH containing each attribute.  If the DUT implements the attribute, the *Status* field will be equal to SUCCESS and the command will contain the requested attribute. If the DUT does not implement the attribute, the *Status* field will not be equal to SUCCESS.  The data type in each command must match the value listed in the specification(s). The data value in each command for the attribute must fall within the valid range described in the specification(s). | |
| 4a | TSTAT.S.A*go*, TSTAT.C.A*go* | TH unicasts a ZCL *write attributes* command frame to DUT to write the respective default value to each optional global attribute of this cluster one at a time. | DUT unicasts a ZCL *write attributes response* command frame to TH for each attribute.  If the attribute is not implemented or the access control of DUT is set to READ, the DUT response will indicate that the attribute write command was not a SUCCESS. If the attribute is implemented and the access control of DUT is set to READ/WRITE, the DUT response will indicate that the write command was a SUCCESS. | |
| 4b | TSTAT.S.A*go*, TSTAT.C.A*go* | TH unicasts a ZCL *read attributes* command frame to DUT to read back each attribute written in step 4a. | DUT unicasts a ZCL *read attributes response* command frame to TH containing the requested attribute.  If the *Status* field of the *write attributes response* command frame was equal to SUCCESS, the updated value is read back. If the *Status* field of the *write attributes response* command frame was not equal to SUCCESS the value is not updated when read back. | |
| *--- End of test case TSTAT-TC-01G ---* | | | |

## Server test cases

### TSTAT-TC-01S: Attributes with server as DUT

This test case verifies the behavior of the attributes of the *thermostat* cluster server.

In this test, the PICS notation TSTAT.S.A*m* represents the list of non-global attributes that are specified as being mandatory. Similarly, the PICS notation TSTAT.S.A*o* represents the list of non-global attributes that are specified as being optional.

#### Scope

General:

* *Read attributes* command (0x00)
* *Read attributes response* command (0x01)
* *Write attributes* command (0x02)
* *Write attributes response* command (0x04)

*Thermostat* cluster (0x0201):

* All non-globalattributes

PICS:

* TSTAT.S,
* TSTAT.S.A*m*, TSTAT.S.A*o*

#### Required devices

|  |  |  |
| --- | --- | --- |
| **Designation** | **Symbol** | **Description** |
| TH CLIENT |  | Test harness client implementing:   * The *thermostat* cluster client. |
| DUT SERVER |  | Device under test server:   * The *thermostat* cluster server. |

#### Initial conditions

|  |  |
| --- | --- |
| **Item** | **Initial Conditions** |
| 1 | A packet sniffer shall be observing the communication over the air interface. |
| 2 | All devices are factory new and powered off until used. |

#### Test preparation

|  |  |
| --- | --- |
| **Add TH CLIENT and DUT SERVER to a ZigBee network** | |
|  |  |
| *Before* | *After* |

| **TSTAT-TC-01S: Attributes with server as DUT** | | |
| --- | --- | --- |
| **Item** | **Preparation Step** | **Observation** |
| P1 | Form a ZigBee network. | Observe appropriate command frame to form the network. |
| P2 | Power on TH CLIENT and DUT SERVER. | TH CLIENT and DUT SERVER are powered on. |
| P3 | Join TH CLIENT and DUT SERVER to a ZigBee network. | Observe appropriate communication between TH CLIENT, DUT SERVER and any other relevant node on the ZigBee network. |
| *--- End of test case TSTAT-TC-01S preparation ---* | | |

#### Test procedure

| **TSTAT-TC-01S: Attributes with server as DUT** | | | | |
| --- | --- | --- | --- | --- |
| **Item** | **PICS** | **Test Harness Step** | **DUT pass Verification** | |
| 1 | TSTAT.S.A*m* | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read each mandatory attribute of this cluster one at a time. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT containing each requested attribute.  The data type in each command must match the value listed in the specification(s). The data value in each command for the attribute must fall within the valid range described in the specification(s). | |
| 2a | TSTAT.S.A*m* | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the respective default value to each mandatory attribute of this cluster one at a time. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT for each attribute.  If the access control of DUT SERVER is set to READ, the DUT SERVER response will indicate that the attribute write command was not a SUCCESS. If the access control of DUT SERVER is set to READ/WRITE, the DUT SERVER response will indicate that the write command was a SUCCESS. | |
| 2b | TSTAT.S.A*m* | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read back each attribute written in step 2a. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT containing the requested attribute.  If the *Status* field of the *write attributes response* command frame was equal to SUCCESS, the updated value is read back. If the *Status* field of the *write attributes response* command frame was not equal to SUCCESS the value is not updated when read back. | |
| *Continued…* | | | | |
| 3 | TSTAT.S.A*o* | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read each optional attribute of this cluster one at a time. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT containing each attribute.  If the DUT SERVER implements the attribute, the *Status* field will be equal to SUCCESS and the command will contain the requested attribute. If the DUT SERVER does not implement the attribute, the *Status* field will not be equal to SUCCESS.  The data type in each command must match the value listed in the specification(s). The data value in each command for the attribute must fall within the valid range described in the specification(s). | |
| 4a | TSTAT.S.A*o* | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the respective default value to each optional attribute of this cluster one at a time. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT for each attribute.  If the attribute is not implemented or the access control of DUT SERVER is set to READ, the DUT SERVER response will indicate that the attribute write command was not a SUCCESS. If the attribute is implemented and the access control of DUT SERVER is set to READ/WRITE, the DUT response will indicate that the write command was a SUCCESS. | |
| *Continued…* | | | | |
| 4b | TSTAT.S.A*o* | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read back each attribute written in step 4a. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT containing the requested attribute.  If the *Status* field of the *write attributes response* command frame was equal to SUCCESS, the updated value is read back. If the *Status* field of the *write attributes response* command frame was not equal to SUCCESS the value is not updated when read back. | |
| *--- End of test case TSTAT-TC-01S ---* | | | |

### TSTAT-TC-02S: Setpoint Test Cases with server as a DUT

This test case verifies the primary functionality of the *thermostat* cluster server in respect to measuring relative humidity changes.

#### Scope

General:

* *Read attributes* command (0x00)
* *Read attributes response* command (0x01)
* *Write attributes* command (0x02)
* *Write attributes response* command (0x03)

*Thermostat* cluster (0x0201):

* *AbsMinHeatSetpointLimit* attribute (0x0003)
* *AbsMaxHeatSetpointLimit* attribute (0x0004)
* *AbsMinCoolSetpointLimit* attribute (0x0005)
* *AbsMaxCoolSetpointLimit* attribute (0x0006)
* *HVACSystemTypeConfiguration* attribute (0x0009)
* *LocalTemperatureCalibration* attribute (0x0010)
* *OccupiedCoolingSetpoint* attribute (0x0011)
* *OccupiedHeatingSetpoint* attribute (0x0012)
* *UnoccupiedCoolingSetpoint* attribute (0x0013)
* *UnoccupiedHeatingSetpoint* attribute (0x0014)
* *MinHeatSetpointLimit* attribute (0x0015)
* *MaxHeatSetpointLimit* attribute (0x0016)
* *MinCoolSetpointLimit* attribute (0x0017)
* *MaxCoolSetpointLimit* attribute (0x0018)
* *MinSetpointDeadBand* attribute (0x0019)
* *RemoteSensing* attribute (0x001A)
* *ControlSequenceOfOperation* attribute (0x001B)
* *SystemMode* attribute (0x001C)
* *Setpoint raise/lower* command (0x00)

PICS:

* TSTAT.S
* TSTAT.S.A0003 - TSTAT.S.A0006, TSTAT.S.A0009, TSTAT.S.A0010 - TSTAT.S.A001C, TSTAT.S.C0.Resp

#### Required devices

|  |  |  |
| --- | --- | --- |
| **Designation** | **Symbol** | **Description** |
| TH CLIENT |  | Test harness client implementing:   * The *thermostat* cluster client. |
| DUT SERVER |  | Device under test server:   * The *thermostat* cluster server. |

#### Initial conditions

|  |  |
| --- | --- |
| **Item** | **Initial Conditions** |
| 1 | A packet sniffer shall be observing the communication over the air interface. |
| 2 | All devices are factory new and powered off until used. |

#### Test preparation

|  |  |
| --- | --- |
| **Add TH CLIENT and DUT SERVER to a ZigBee network** | |
|  |  |
| *Before* | *After* |

| **TSTAT-TC-02S: Setpoint Test Cases with server as DUT** | | |
| --- | --- | --- |
| **Item** | **Preparation Step** | **Observation** |
| P1 | Form a ZigBee network. | Observe appropriate command frame to form the network. |
| P2 | Power on TH CLIENT and DUT SERVER. | TH CLIENT and DUT SERVER are powered on. |
| P3 | Join TH CLIENT and DUT SERVER to a ZigBee network. | Observe appropriate communication between TH CLIENT, DUT SERVER and any other relevant node on the ZigBee network. |
| *--- End of test case TSTAT-TC-02S preparation ---* | | |

#### Test procedure

| **TSTAT-TC-02S: Setpoint Test Cases with server as DUT** | | | | |
| --- | --- | --- | --- | --- |
| **Item** | **PICS** | **Test Harness Step** | **DUT pass Verification** | |
| Setting *LocalTemperatureCalibration* | | | | |
| 1a | TSTAT.S.A0010 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *LocalTemperatureCalibration* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *LocalTemperatureCalibration* attribute has a value in the range 0xE7 – 0x19. | |
| 1b | TSTAT.S.A0010 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *LocalTemperatureCalibration* attribute to a different but valid value from the range 0xE7 – 0x19. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 1c | TSTAT.S.A0010 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *LocalTemperatureCalibration* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *LocalTemperatureCalibration* attribute has the updated value. | |
| 1d | TSTAT.S.A0010 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *LocalTemperatureCalibration* attribute to a value above the valid range. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0xC4 (LIMIT\_REACHED). | |
| 1e | TSTAT.S.A0010 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *LocalTemperatureCalibration* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *LocalTemperatureCalibration* attribute has the maximum valid value of 0x19 (25). | |
| 1f | TSTAT.S.A0010 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *LocalTemperatureCalibration* attribute to a value below the valid range. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0xC4 (LIMIT\_REACHED). | |
| 1g | TSTAT.S.A0010 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *LocalTemperatureCalibration* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *LocalTemperatureCalibration* attribute has the minimum value of 0xE7 (-25). | |
| 1h | TSTAT.S.A0010 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *LocalTemperatureCalibration* attribute to a value 0xE7. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 1i | TSTAT.S.A0010 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *LocalTemperatureCalibration* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *LocalTemperatureCalibration* attribute has the value 0xE7. | |
| 1j | TSTAT.S.A0010 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *LocalTemperatureCalibration* attribute to a value 0x19. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 1k | TSTAT.S.A0010 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *LocalTemperatureCalibration* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *LocalTemperatureCalibration* attribute has the value 0x19. | |
| Setting *OccupiedCoolingSetpoint / OccupiedHeatingSetpoint* | | | | |
| 2a | TSTAT.S.A0005 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *AbsMinCoolSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *AbsMinCoolSetpointLimit* attribute has a value in the range 0x954d – 0x7fff. | |
| 2b | TSTAT.S.A0006 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *AbsMaxCoolSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *AbsMaxCoolSetpointLimit* attribute has a value in the range 0x954d – 0x7fff and *AbsMaxCoolSetpointLimit* is greater than *AbsMinCoolSetpointLimit*. | |
| 2c | TSTAT.S.A0017 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MinCoolSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MinCoolSetpointLimit* attribute has a value in the range *AbsMinCoolSetpointLimit* – *AbsMaxCoolSetpointLimit* as read in steps 2a and 2b (or their default values if unsupported). | |
| 2d | TSTAT.S.A0018 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MaxCoolSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MaxCoolSetpointLimit* attribute has a value in the range *AbsMinCoolSetpointLimit* – *AbsMaxCoolSetpointLimit* as read in steps 2a and 2b and *MaxCoolSetpointLimit* is greater than *MinCoolSetpointLimit* as read in 2c. | |
| 2e | TSTAT.S.A0011 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *OccupiedCoolingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *OccupiedCoolingSetpoint* attribute has a value in the range *MinCoolSetpointLimit* – *MaxCoolSetpointLimit* as read in step 2c and 2d (or their default values if unsupported). | |
| 2f | TSTAT.S.A0003 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *AbsMinHeatSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *AbsMinHeatSetpointLimit* attribute has a value in the range 0x954d – 0x7fff. | |
| 2g | TSTAT.S.A0004 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *AbsMaxHeatSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *AbsMaxHeatSetpointLimit* attribute has a value in the range 0x954d – 0x7fff. | |
| 2h | TSTAT.S.A0015 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MinHeatSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MinHeatSetpointLimit* attribute has a value in the range *AbsMinHeatSetpointLimit* – *AbsMaxHeatSetpointLimit* as read in steps 2f and 2g (or their default values if unsupported). | |
| 2i | TSTAT.S.A0016 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MaxHeatSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MaxHeatSetpointLimit* attribute has a value in the range *AbsMinHeatSetpointLimit* – *AbsMaxHeatSetpointLimit* as read in steps 2f and 2g and *MaxHeatSetpointLimit* is greater than *MinHeatSetpointLimit* as read in 2h. | |
| 2j | TSTAT.S.A0019 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MinSetpointDeadBand* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MinSetpointDeadBand* attribute has a value in the range 0x0a – 0x19. | |
| 2k | TSTAT.S.A0012 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *OccupiedHeatingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *OccupiedHeatingSetpoint* attribute has a value in the range *MinHeatSetpointLimit* – *MaxHeatSetpointLimit* as read in step 2h and 2i (or their default values if unsupported), and lower than the *OccupiedCoolingSetpoint* by at least the value of the *MinSetpointDeadBand* as read in step 2j.  **Note:** If the optional *MinSetpointDeadBand* attribute is not supported, the default value of 0x19 (2.5°C) is used. | |
| 2l | TSTAT.S.A0011 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *OccupiedCoolingSetpoint* attribute to a value above *MaxCoolSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x87 (INVALID\_VALUE). | |
| 2m | TSTAT.S.A0011 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *OccupiedCoolingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *OccupiedCoolingSetpoint* attribute has the unmodified value from step 2e. | |
| 2n | TSTAT.S.A0011 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *OccupiedCoolingSetpoint* attribute to a value below the *MinCoolSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x87 (INVALID\_VALUE). | |
| 2o | TSTAT.S.A0011 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *OccupiedCoolingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *OccupiedCoolingSetpoint* attribute has the unmodified value from step 2e. | |
| 2p | TSTAT.S.A0012 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *OccupiedHeatingSetpoint* attribute to a value above *MaxHeatSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x87 (INVALID\_VALUE). | |
| 2q | TSTAT.S.A0012 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *OccupiedHeatingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *OccupiedHeatingSetpoint* attribute has the unmodified value from step 2k. | |
| 2r | TSTAT.S.A0012 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *OccupiedHeatingSetpoint* attribute to a value below the *MinHeatSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x87 (INVALID\_VALUE). | |
| 2s | TSTAT.S.A0012 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *OccupiedHeatingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *OccupiedHeatingSetpoint* attribute has the unmodified value from step 2k. | |
| 3a | TSTAT.S.A0011 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *OccupiedCoolingSetpoint* attribute to a value *MaxCoolSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 3b | TSTAT.S.A0011 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *OccupiedCoolingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *OccupiedCoolingSetpoint* attribute has the value *MaxCoolSetpointLimit*. | |
| 3c | TSTAT.S.A0012 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *OccupiedHeatingSetpoint* attribute to a value *MinHeatSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 3d | TSTAT.S.A0012 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *OccupiedHeatingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *OccupiedHeatingSetpoint* attribute has the value *MinHeatSetpointLimit*. | |
| 3e | TSTAT.S.A0011 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *OccupiedCoolingSetpoint* attribute to a value which is max(*MinCoolSetpointLimit,* (*MinHeatSetpointLimit + MinSetpointDeadBand)*).  **Note:** If the optional *MinSetpointDeadBand* attribute is not supported, the default value of 0x19 (2.5°C) is used | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 3f | TSTAT.S.A0011 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *OccupiedCoolingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *OccupiedCoolingSetpoint* attribute has the updated value from step 3e. | |
| 3g | TSTAT.S.A0011 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *OccupiedCoolingSetpoint* attribute to a value *MaxCoolSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 3h | TSTAT.S.A0011 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *OccupiedCoolingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *OccupiedCoolingSetpoint* attribute has the value *MaxCoolSetpointLimit*. | |
| 3i | TSTAT.S.A0012 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *OccupiedHeatingSetpoint* attribute to a value which is min(*MaxHeatSetpointLimit,* (*MaxCoolSetpointLimit - MinSetpointDeadBand)*).  **Note**: If the optional *MinSetpointDeadBand* attribute is not supported, the default value of 0x19 (2.5°C) is used | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 3j | TSTAT.S.A0012 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *OccupiedHeatingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *OccupiedHeatingSetpoint* attribute has the updated value from step 3i. | |
| 3k | TSTAT.S.A0012 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *OccupiedHeatingSetpoint* attribute to a value *MinHeatSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 3l | TSTAT.S.A0012 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *OccupiedHeatingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *OccupiedHeatingSetpoint* attribute has the value *MinHeatSetpointLimit*. | |
| 3m | TSTAT.S.A0011 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *OccupiedCoolingSetpoint* attribute to a value from the mid-part of the range *MinCoolSetpointLimit* – *MaxCoolSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 3n | TSTAT.S.A0011 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *OccupiedCoolingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *OccupiedCoolingSetpoint* attribute has the updated value from the step 3m. | |
| 3o | TSTAT.S.A0012 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *OccupiedHeatingSetpoint* attribute to a value from the mid-part of the range *MinHeatSetpointLimit* – *MaxHeatSetpointLimit.*  **Note**:  If *OccupiedCoolingSetpoint* is supported, the value for *OccupiedHeatingSetpoint* shall be chosen in such a way that the difference between *OccupiedHeatingSetpoint* and *OccupiedCoolingSetpoint* is larger than or equals to the *MinSetpointDeadBand* value specified at step 3a.  If the optional *MinSetpointDeadBand* attribute is not supported, the default value of 0x19 (2.5°C) is used | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 3p | TSTAT.S.A0012 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *OccupiedHeatingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *OccupiedHeatingSetpoint* attribute has the updated value from the step 3o. | |
| Setting *UnoccupiedCoolingSetpoint / UnoccupiHeatingSetpoint*  ***Test step conditional on the optional* UnoccupiedCoolingSetpoint and UnoccupieHeatingSetpoint *attribute being supported*** | | | | |
| 4a | TSTAT.S.A0013 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *UnoccupiedCoolingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *UnoccupiedCoolingSetpoint* attribute has a value in the range *MinCoolSetpointLimit* – *MaxCoolSetpointLimit* as read in step 2c and 2d (or their default values if unsupported). | |
| 4b | TSTAT.S.A0013 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *UnoccupiedCoolingSetpoint* attribute to a value above *MaxCoolSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x87 (INVALID\_VALUE). | |
| 4c | TSTAT.S.A0013 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *UnoccupiedCoolingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *UnoccupiedCoolingSetpoint* attribute has the unmodified value from step 4a. | |
| 4d | TSTAT.S.A0013 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *UnoccupiedCoolingSetpoint* attribute to a value below the *MinCoolSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x87 (INVALID\_VALUE). | |
| 4e | TSTAT.S.A0013 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *UnoccupiedCoolingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *UnoccupiedCoolingSetpoint* attribute has the unmodified value from step 4a. | |
| 4f | TSTAT.S.A0014 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *UnoccupiedHeatingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *UnoccupiedHeatingSetpoint* attribute has a value in the range *MinHeatSetpointLimit* – *MaxHeatSetpointLimit* as read in step 2h and 2i (or their default values if unsupported), and lower than the *UnoccupiedCoolingSetpoint* by at least the value of the *MinSetpointDeadBand* as read in step 4a.  **Note:** If the optional *MinSetpointDeadBand* attribute is not supported, the default value of 0x19 (2.5°C) is used. | |
| 4g | TSTAT.S.A0014 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *UnoccupiedHeatingSetpoint* attribute to a value above *MaxHeatSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x87 (INVALID\_VALUE). | |
| 4h | TSTAT.S.A0014 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *UnoccupiedHeatingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *UnoccupiedHeatingSetpoint* attribute has the unmodified value from step 4f. | |
| 4i | TSTAT.S.A0014 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *UnoccupiedHeatingSetpoint* attribute to a value below the *MinHeatSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x87 (INVALID\_VALUE). | |
| 4j | TSTAT.S.A0014 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *UnoccupiedHeatingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *UnoccupiedHeatingSetpoint* attribute has the unmodified value from step 4f. | |
| 5a | TSTAT.S.A0013 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *UnoccupiedCoolingSetpoint* attribute to a value *MaxCoolSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 5b | TSTAT.S.A0013 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *UnoccupiedCoolingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *UnoccupiedCoolingSetpoint* attribute has the value *MaxCoolSetpointLimit*. | |
| 5c | TSTAT.S.A0014 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *UnoccupiedHeatingSetpoint* attribute to a value *MinHeatSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 5d | TSTAT.S.A0014 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *UnoccupiedHeatingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *UnoccupiedHeatingSetpoint* attribute has the value *MinHeatSetpointLimit*. | |
| 5e | TSTAT.S.A0013 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *UnoccupiedCoolingSetpoint* attribute to a value which is max(*MinCoolSetpointLimit,* (*MinHeatSetpointLimit + MinSetpointDeadBand)*).  **Note:** If the optional *MinSetpointDeadBand* attribute is not supported, the default value of 0x19 (2.5°C) is used | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 5f | TSTAT.S.A0013 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *UnoccupiedCoolingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *UnoccupiedCoolingSetpoint* attribute has the updated value from step 5e. | |
| 5g | TSTAT.S.A0013 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *UnoccupiedCoolingSetpoint* attribute to a value *MaxCoolSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 5h | TSTAT.S.A0013 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *UnoccupiedCoolingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *UnoccupiedCoolingSetpoint* attribute has the value *MaxCoolSetpointLimit*. | |
| 5i | TSTAT.S.A0014 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *UnoccupiedHeatingSetpoint* attribute to a value which is min(*MaxHeatSetpointLimit,* (*MaxCoolSetpointLimit - MinSetpointDeadBand)*).  **Note**: If the optional *MinSetpointDeadBand* attribute is not supported, the default value of 0x19 (2.5°C) is used | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 5j | TSTAT.S.A0014 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *UnoccupiedHeatingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *UnoccupiedHeatingSetpoint* attribute has the updated value from step 3i. | |
| 5k | TSTAT.S.A0014 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *UnoccupiedHeatingSetpoint* attribute to a value *MinHeatSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 5l | TSTAT.S.A0014 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *UnoccupiedHeatingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *UnoccupiedHeatingSetpoint* attribute has the value *MinHeatSetpointLimit*. | |
| 5m | TSTAT.S.A0013 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *UnoccupiedCoolingSetpoint* attribute to a value from the mid-part of the range *MinCoolSetpointLimit* – *MaxCoolSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 5n | TSTAT.S.A0013 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *UnoccupiedCoolingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *UnoccupiedCoolingSetpoint* attribute has the updated value from the step 3m. | |
| 5o | TSTAT.S.A0014 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *UnoccupiedHeatingSetpoint* attribute to a value from the mid-part of the range *MinHeatSetpointLimit* – *MaxHeatSetpointLimit.*  **Note**:  If *UnoccupiedCoolingSetpoint* is supported, the value for *UnoccupiedHeatingSetpoint* shall be chosen in such a way that the difference between *UnoccupiedHeatingSetpoint* and *UnoccupiedCoolingSetpoint* is larger than or equals to the *MinSetpointDeadBand* value specified at step 3a.  If the optional *MinSetpointDeadBand* attribute is not supported, the default value of 0x19 (2.5°C) is used | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 5p | TSTAT.S.A0014 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *UnoccupiedHeatingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *UnoccupiedHeatingSetpoint* attribute has the updated value from the step 5o. | |
| Setting *MinHeatSetpointLimit*  ***Test step conditional on the optional* MinHeatSetpointLimit *attribute being supported*** | | | | |
| 6a | TSTAT.S.A0015 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MinHeatSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MinHeatSetpointLimit* attribute has a value in the range *AbsMinHeatSetpointLimit* – *AbsMaxHeatSetpointLimit* as read in steps 3a and 3b. | |
| 6b | TSTAT.S.A0015 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *MinHeatSetpointLimit* attribute to a different but valid value from the range *AbsMinHeatSetpointLimit* – *AbsMaxHeatSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 6c | TSTAT.S.A0015 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MinHeatSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MinHeatSetpointLimit* attribute has the updated value. | |
| 6d | TSTAT.S.A0015 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *MinHeatSetpointLimit* attribute to a value above *AbsMaxHeatSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x87 (INVALID\_VALUE). | |
| 6e | TSTAT.S.A0015 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MinHeatSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MinHeatSetpointLimit* attribute has the unmodified value from step 6c. | |
| 6f | TSTAT.S.A0015 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *MinHeatSetpointLimit* attribute to a value below *AbsMinHeatSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x87 (INVALID\_VALUE). | |
| 6g | TSTAT.S.A0015 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MinHeatSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MinHeatSetpointLimit* attribute has the unmodified value from step 6c. | |
| 6h | TSTAT.S.A0015 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *MinHeatSetpointLimit* attribute to a value of *AbsMaxHeatSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 6i | TSTAT.S.A0015 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MinHeatSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MinHeatSetpointLimit* attribute has the value *AbsMaxHeatSetpointLimit*. | |
| 6j | TSTAT.S.A0015 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *MinHeatSetpointLimit* attribute to a value *AbsMinHeatSetpoint*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 6k | TSTAT.S.A0015 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MinHeatSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MinHeatSetpointLimit* attribute has the value *AbsMinHeatSetpoint*. | |
| Setting *MaxHeatSetpointLimit*  ***Test step conditional on the optional* MaxHeatSetpointLimit *attribute being supported*** | | | | |
| 7a | TSTAT.S.A0016 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MaxHeatSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MaxHeatSetpointLimit* attribute has a value in the range *AbsMinHeatSetpointLimit* – *AbsMaxHeatSetpointLimit* as read in steps 3a and 3c. | |
| 7b | TSTAT.S.A0016 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *MaxHeatSetpointLimit* attribute to a different but valid value from the range *AbsMinHeatSetpointLimit* – *AbsMaxHeatSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 7c | TSTAT.S.A0016 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MaxHeatSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MaxHeatSetpointLimit* attribute has the updated value. | |
| 7d | TSTAT.S.A0016 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *MaxHeatSetpointLimit* attribute to a value above *AbsMaxHeatSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x87 (INVALID\_VALUE). | |
| 7e | TSTAT.S.A0016 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MaxHeatSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MaxHeatSetpointLimit* attribute has the unmodified value from step 7c. | |
| 7f | TSTAT.S.A0016 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *MaxHeatSetpointLimit* attribute to a value below *AbsMinHeatSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x87 (INVALID\_VALUE). | |
| 7g | TSTAT.S.A0016 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MaxHeatSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MaxHeatSetpointLimit* attribute has the unmodified value from step 7c. | |
| 7h | TSTAT.S.A0016 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *MaxHeatSetpointLimit* attribute to the value *AbsMinHeatSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 7i | TSTAT.S.A0016 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MaxHeatSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MaxHeatSetpointLimit* attribute has the value *AbsMinHeatSetpointLimit*. | |
| 7j | TSTAT.S.A0016 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *MaxHeatSetpointLimit* attribute to the value *AbsMaxHeatSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 7k | TSTAT.S.A0016 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MaxHeatSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MaxHeatSetpointLimit* attribute has the value *AbsMaxHeatSetpointLimit*. | |
| Setting *MinCoolSetpointLimit*  ***Test step conditional on the optional* MinCoolSetpointLimit *attribute being supported*** | | | | |
| 8a | TSTAT.S.A0017 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MinCoolSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MinCoolSetpointLimit* attribute has a value in the range *AbsMinCoolSetpointLimit* – *AbsMaxCoolSetpointLimit* as read in step 3a and 3b. | |
| 8b | TSTAT.S.A0017 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *MinCoolSetpointLimit* attribute to a different but valid value from the range *AbsMinCoolSetpointLimit* – *AbsMaxCoolSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 8c | TSTAT.S.A0017 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MinCoolSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MinCoolSetpointLimit* attribute has the updated value. | |
| 8d | TSTAT.S.A0017 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *MinCoolSetpointLimit* attribute to a value above *AbsMaxCoolSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x87 (INVALID\_VALUE). | |
| 8e | TSTAT.S.A0017 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MinCoolSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MinCoolSetpointLimit* attribute has the unmodified value from step 8c. | |
| 8f | TSTAT.S.A0017 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *MinCoolSetpointLimit* attribute to a value below *AbsMinCoolSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x87 (INVALID\_VALUE). | |
| 8g | TSTAT.S.A0017 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MinCoolSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MinCoolSetpointLimit* attribute has the unmodified value from step 8c. | |
| 8h | TSTAT.S.A0017 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *MinCoolSetpointLimit* attribute to the value  *AbsMaxCoolSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 8i | TSTAT.S.A0017 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MinCoolSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MinCoolSetpointLimit* attribute has the value *AbsMaxCoolSetpointLimit*. | |
| 8j | TSTAT.S.A0017 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *MinCoolSetpointLimit* attribute to a value *AbsMinCoolSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 8k | TSTAT.S.A0017 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MinCoolSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MinCoolSetpointLimit* attribute has the value *AbsMinnCoolSetpointLimit*. | |
| Setting *MaxCoolSetpointLimit*  ***Test step conditional on the optional* MaxCoolSetpointLimit *attribute being supported*** | | | | |
| 9a | TSTAT.S.A0018 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MaxCoolSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MaxCoolSetpointLimit* attribute has a value in the range *AbsMinCoolSetpointLimit* – *AbsMaxCoolSetpointLimit* as read in step 3a and 3b. | |
| 9b | TSTAT.S.A0018 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *MaxCoolSetpointLimit* attribute to a different but valid value from the range *AbsMinCoolSetpointLimit* – *AbsMaxCoolSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 9c | TSTAT.S.A0018 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MaxCoolSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MaxCoolSetpointLimit* attribute has the updated value. | |
| 9d | TSTAT.S.A0018 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *MaxCoolSetpointLimit* attribute to a value above *AbsMaxCoolSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x87 (INVALID\_VALUE). | |
| 9e | TSTAT.S.A0018 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MaxCoolSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MaxCoolSetpointLimit* attribute has the unmodified value from step 9c. | |
| 9f | TSTAT.S.A0018 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *MaxCoolSetpointLimit* attribute to a value below *AbsMinCoolSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x87 (INVALID\_VALUE). | |
| 9g | TSTAT.S.A0018 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MaxCoolSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MaxCoolSetpointLimit* attribute has the unmodified value from step 9c. | |
| 9h | TSTAT.S.A0018 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *MaxCoolSetpointLimit* attribute to a value *AbsMinCoolSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 9i | TSTAT.S.A0018 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MaxCoolSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MaxCoolSetpointLimit* attribute has the value *AbsMinCoolSetpointLimit*. | |
| 9j | TSTAT.S.A0018 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *MaxCoolSetpointLimit* attribute to a value *AbsMaxCoolSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 9k | TSTAT.S.A0018 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MaxCoolSetpointLimit* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MaxCoolSetpointLimit* attribute has the value *AbsMaxCoolSetpointLimit*. | |
| Setting *MinSetpointDeadBand*  ***Test step conditional on the optional* MinSetpointDeadBand *attribute being supported*** | | | | |
| 10a | TSTAT.S.A0019 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MinSetpointDeadBand* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MinSetpointDeadBand* attribute has a value in the range 0x0a – 0x19. | |
| 10b | TSTAT.S.A0019 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *MinSetpointDeadBand* attribute to a different but valid value from the range 0x0a – 0x19. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 10c | TSTAT.S.A0019 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MinSetpointDeadBand* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MinSetpointDeadBand* attribute has the updated value. | |
| 10d | TSTAT.S.A0019 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *MinSetpointDeadBand* attribute to a value above 0x19. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x87 (INVALID\_VALUE). | |
| 10e | TSTAT.S.A0019 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MinSetpointDeadBand* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MinSetpointDeadBand* attribute has the unmodified value from step 10c. | |
| 10f | TSTAT.S.A0019 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *MinSetpointDeadBand* attribute to a value below 0x0a. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x87 (INVALID\_VALUE). | |
| 10g | TSTAT.S.A0019 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MinSetpointDeadBand* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MinSetpointDeadBand* attribute has the unmodified value from step 10c. | |
| 10h | TSTAT.S.A0019 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *MinSetpointDeadBand* attribute to a value 0x19. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 10i | TSTAT.S.A0019 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MinSetpointDeadBand* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MinSetpointDeadBand* attribute has the value 0x19. | |
| 10j | TSTAT.S.A0019 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *MinSetpointDeadBand* attribute to a value 0x0a. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 10k | TSTAT.S.A0019 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *MinSetpointDeadBand* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *MinSetpointDeadBand* attribute has the value 0x0a. | |
| Setting *RemoteSensing*  ***Test step conditional on the optional* RemoteSensing *attribute being supported*** | | | | |
| 11a | TSTAT.S.A001A | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *RemoteSensing* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *RemoteSensing* attribute has a value lower than 0x08. | |
| 11b | TSTAT.S.A001A | TH CLIENT XORs the 3lsb of the received value of the *RemoteSensing* attribute.  TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *RemoteSensing* attribute to a value being the result of the XOR operation above. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 11c | TSTAT.S.A001A | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *RemoteSensing* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *RemoteSensing* attribute has the updated value. | |
| 11d | TSTAT.S.A001A | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *RemoteSensing* attribute to a value 0x0f. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x87 (INVALID\_VALUE). | |
| 11e | TSTAT.S.A001A | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *RemoteSensing* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *RemoteSensing* attribute has the value as read at step 11c. | |
| 11f | TSTAT.S.A001A | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *RemoteSensing* attribute to a value 0x07. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 11g | TSTAT.S.A001A | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *RemoteSensing* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *RemoteSensing* attribute the value 0x07. | |
| 11h | TSTAT.S.A001A | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *RemoteSensing* attribute to a value 0x00. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 11i | TSTAT.S.A001A | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *RemoteSensing* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *RemoteSensing* attribute the value 0x00. | |
| Setting *ControlSequenceOfOperation* | | | | |
| 12a | TSTAT.S.A001B | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *ControlSequenceOfOperation* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *ControlSequenceOfOperation* attribute has a value lower than 0x06. | |
| 12b | TSTAT.S.A001B | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *ControlSequenceOfOperation* attribute to a different but valid value TSTAT.PIXIT.CONTROLSEQUENCE from the range 0x00 – 0x05. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 12c | TSTAT.S.A001B | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *ControlSequenceOfOperation* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *ControlSequenceOfOperation* attribute has the updated value. | |
| 12d | TSTAT.S.A001B | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *ControlSequenceOfOperation* attribute to a value 0x0f. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x87 (INVALID\_VALUE). | |
| 12e | TSTAT.S.A001B | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *ControlSequenceOfOperation* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *ControlSequenceOfOperation* attribute has the value as read at step 12c. | |
| Setting *SystemMode*  ***Consider PIXIT item as not all of the 10 system modes will be supported by each DUT*** | | | | |
| 13a | TSTAT.S.A001C | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *SystemMode* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *SystemMode* attribute has a value lower than 0x0A. | |
| 13b | TSTAT.S.A001C | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *SystemMode* attribute to each of the possible values within the valid range 0x00 – 0x09 in turn.  After each write action, TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *SystemMode* attribute. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT.  If the DUT supports the written value, the *status* field of the ZCL *write attributes response* equal to 0x00 (SUCCESS), and the *SystemMode* value in the ZCL *read attributes* command is set to the updated value.  If the DUT does NOT support the written value, the *status* field of the ZCL *write attributes response* equal to 0x87 (INVALID\_VALUE), and the *SystemMode* value in the ZCL *read attributes* command is unmodified. | |
| 13c | TSTAT.S.A001C | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *SystemMode* attribute a value 0x0A. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x87 (INVALID\_VALUE). | |
| 13d | TSTAT.S.A001C | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *SystemMode* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *SystemMode* attribute has its last value from step 13b. | |
| *Setpoint raise/lower* command | | | | |
| 14a | TSTAT.S.A0011 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *OccupiedCoolingSetpoint* attribute to a value *MaxCoolSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 14b | TSTAT.S.A0011 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *OccupiedCoolingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *OccupiedCoolingSetpoint* attribute has the value *MaxCoolSetpointLimit*. | |
| 14c | TSTAT.S.A0012 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *OccupiedHeatingSetpoint* attribute to its default value 0x07d0 (20°C). | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 14d | TSTAT.S.C00.Resp | TH CLIENT unicasts a *setpoint raise/lower* command frame of the *thermostat* cluster to DUT SERVER with:  - *mode* field set to Heat (0x00), - *amount* field set to 0xE2 (-30 units = -3 degrees). | DUT SERVER unicasts a ZCL *default response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 14e | TSTAT.S.A0012 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *OccupiedHeatingSetpoint*. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT; the *OccupiedHeatingSetpoint* attribute has the updated value 0x06A4 (17°C). | |
| 15a | TSTAT.S.A0012 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *OccupiedHeatingSetpoint* attribute to its default value 0x07d0 (20°C). | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 15b | TSTAT.S.C00.Resp | TH CLIENT unicasts a *setpoint raise/lower* command frame of the *thermostat* cluster to DUT SERVER with:  - *mode* field set to Heat (0x00), - *amount* field set to 0x1E (+30 units = +3 degrees). | DUT SERVER unicasts a ZCL *default response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 15c | TSTAT.S.A0012 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *OccupiedHeatingSetpoint*. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT; the *OccupiedHeatingSetpoint* attribute has the updated value 0x08fc (23°C). | |
| 16a | TSTAT.S.A0012 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *OccupiedHeatingSetpoint* attribute to a value *MinHeatSetpointLimit*. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 16b | TSTAT.S.A0012 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *OccupiedHeatingSetpoint* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT. *OccupiedHeatingSetpoint* attribute has the value *MinHeatSetpointLimit*. | |
| 16c | TSTAT.S.A0011 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *OccupiedCoolingSetpoint* attribute to its default value 0x0a28 (26°C). | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 16d | TSTAT.S.C00.Resp | TH CLIENT unicasts a *setpoint raise/lower* command frame of the *thermostat* cluster to DUT SERVER with:  - *mode* field set to Cool (0x01), - *amount* field set to 0xE2 (-30 units = -3 degrees). | DUT SERVER unicasts a ZCL *default response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 16e | TSTAT.S.A0011 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *OccupiedCoolingSetpoint*. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT; the *OccupiedCoolingSetpoint* attribute has the updated value 0x08fc (23°C). | |
| 17a | TSTAT.S.A0011 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *OccupiedCoolingSetpoint* attribute to its default value 0x0a28 (26°C). | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 17b | TSTAT.S.C00.Resp | TH CLIENT unicasts a *setpoint raise/lower* command frame of the *thermostat* cluster to DUT SERVER with:  - *mode* field set to Cool (0x01), - *amount* field set to 0x1E (+30 units = +3 degrees). | DUT SERVER unicasts a ZCL *default response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 17c | TSTAT.S.A0011 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *OccupiedCoolingSetpoint*. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT; the *OccupiedCoolingSetpoint* attribute has the updated value 0x0b54 (29°C). | |
| 18a | TSTAT.S.A0011 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *OccupiedCoolingSetpoint* attribute to its default value 0x0a28 (26°C). | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 18b | TSTAT.S.A0012 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *OccupiedHeatingSetpoint* attribute to its default value 0x07d0 (20°C). | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 18c | TSTAT.S.C00.Resp | TH CLIENT unicasts a *setpoint raise/lower* command frame of the *thermostat* cluster to DUT SERVER with:  - *mode* field set to Both (0x02), - *amount* field set to 0xE2 (-30 units = -3 degrees). | DUT SERVER unicasts a ZCL *default response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 18d | TSTAT.S.A0011 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *OccupiedCoolingSetpoint*. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT; the *OccupiedCoolingSetpoint* attribute has the updated value 0x08fc (23°C). | |
| 18e | TSTAT.S.A0012 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *OccupiedHeatingSetpoint*. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT; the *OccupiedHeatingSetpoint* attribute has the updated value 0x06A4 (17°C). | |
| 19a | TSTAT.S.A0011 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *OccupiedCoolingSetpoint* attribute to its default value 0x0a28 (26°C). | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 19b | TSTAT.S.A0012 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *OccupiedHeatingSetpoint* attribute to its default value 0x07d0 (20°C). | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 19c | TSTAT.S.C00.Resp | TH CLIENT unicasts a *setpoint raise/lower* command frame of the *thermostat* cluster to DUT SERVER with:  - *mode* field set to Both (0x02), - *amount* field set to 0x1E (+30 units = +3 degrees). | DUT SERVER unicasts a ZCL *default response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 19d | TSTAT.S.A0012 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *OccupiedHeatingSetpoint*. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT; the *OccupiedHeatingSetpoint* attribute has the updated value 0x08fc (23°C). | |
| 19e | TSTAT.S.A0011 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *OccupiedCoolingSetpoint*. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT; the *OccupiedCoolingSetpoint* attribute has the updated value 0x0b54 (29°C). | |
| *HVACSystemTypeConfiguration* attribute | | | | |
| 20a | TSTAT.S.A0009 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *HVACSystemTypeConfiguration* attribute to a value 0b00000100 (Heat Stage 2). | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 20b | TSTAT.S.A0009 | TH CLIENT unicasts a ZCL *read attributes* command frame to DUT SERVER to read the *HVACSystemTypeConfiguration* attribute. | DUT SERVER unicasts a ZCL *read attributes response* command; the *HVACSystemTypeConfiguration* attribute has a value 0b00000100 (Heat Stage 2). | |
| *--- End of test case TSTAT-TC-02S ---* | | | |

### TSTAT-TC-03S: Schedule test cases with server as a DUT

This test case verifies the primary functionality of the *thermostat* cluster server in respect to storing and updating schedules.

#### Scope

General:

* *Write attributes* command (0x02)
* *Write attributes response* command (0x03)

*Thermostat* cluster (0x0201):

* *TemperatureSetpointHold* attribute (0x0023)
* *TemperatureSetpointHoldDuration* attribute (0x0024)
* *set weekly schedule* command (0x01)
* *get weekly* *schedule* command (0x02)
* *get weekly schedule response* command (0x00)
* *clear weekly schedule* command (0x03)
* *get relay status log* command (0x04)
* *get relay status log response* command (0x01)

PICS:

* TSTAT.S
* TSTAT.S.A0023, TSTAT.S.A0024,
* TSTAT.S.C01.Rsp, TSTAT.S.C02.Rsp, TSTAT.S.C03.Rsp, TSTAT.S.C04.Rsp, TSTAT.S.C00.Tx, TSTAT.S.C01.Tx

#### Required devices

|  |  |  |
| --- | --- | --- |
| **Designation** | **Symbol** | **Description** |
| TH CLIENT |  | Test harness client implementing:   * The *thermostat* cluster client. |
| DUT SERVER |  | Device under test server:   * The *thermostat* cluster server. |

#### Initial conditions

|  |  |
| --- | --- |
| **Item** | **Initial Conditions** |
| 1 | A packet sniffer shall be observing the communication over the air interface. |
| 2 | All devices are factory new and powered off until used. |

#### Test preparation

|  |  |
| --- | --- |
| **Add TH CLIENT and DUT SERVER to a ZigBee network** | |
|  |  |
| *Before* | *After* |

| **TSTAT-TC-03S: Schedule test cases with server as a DUT** | | |
| --- | --- | --- |
| **Item** | **Preparation Step** | **Observation** |
| P1 | Form a ZigBee network. | Observe appropriate command frame to form the network. |
| P2 | Power on TH CLIENT and DUT SERVER. | TH CLIENT and DUT SERVER are powered on. |
| P3 | Join TH CLIENT and DUT SERVER to a ZigBee network. | Observe appropriate communication between TH CLIENT, DUT SERVER and any other relevant node on the ZigBee network. |
| *--- End of test case TSTAT-TC-03S preparation ---* | | |

#### Test procedure

| **TSTAT-TC-03S: Schedule test cases with server as a DUT** | | | | |
| --- | --- | --- | --- | --- |
| **Item** | **PICS** | **Test Harness Step** | **DUT pass Verification** | |
| 1 | TSTAT.S.A0023 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *TemperatureSetpointHold* attribute to value 0x01, enabling temperature setpoint hold. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 2a | TSTAT.S.A0023  TSTAT.S.A0024 | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER to write the *TemperatureSetpointHoldDuration* attribute to value 0x0001. | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 2b | TSTAT.S.A0023  TSTAT.S.A0024 | - | Verify that the *TemperatureSetpointHold* lasts for 1 minute after the DUT receives the updated *TemperatureSetpointHoldDuration* value. | |
| 3 | TSTAT.S.C01.Rsp | TH CLIENT unicasts a *set weekly schedule* command frame of the *thermostat* cluster to DUT SERVER, correctly formatted and carrying correct parameter values, with:  - *Day of week for sequence* set to the current week day;  - *Mode for Sequence* field set to 0x01 (Heat mode only). | DUT SERVER unicasts a ZCL *default response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS).  If the *set weekly schedule* command frame is not accepted by the DUT SERVER, the DUT SERVER should unicasts a ZCL default response command frame to TH CLIENT with the status field equal to the appropriate value: 0x89 (INSUFFICIENT\_SPACE) if the total number of transitions  sent is greater than what the DUT SERVER supports; 0x87 (INVALID\_VALUE) if any of the set points sent in the entire sequence is out of range of w.r.t. DUT SERVER’s absolute min/max setpoint limit. | |
| 4 | TSTAT.S.C02.Rsp  TSTAT.S.C00.Tx | TH CLIENT unicasts a *get weekly schedule* command frame of the *thermostat* cluster to DUT SERVER, correctly formatted with:  - *Days to return* set to the current week day;  - *Mode to return* set to 0x01 (Heat mode only). | DUT SERVER unicasts a ZCL *get weekly schedule response* command frame to TH CLIENT, carrying the data as in the *set weekly schedule* command of step 3. | |
| 5 | TSTAT.S.C01.Rsp | TH CLIENT unicasts a *set weekly schedule* command frame of the *thermostat* cluster to DUT SERVER, correctly formatted and carrying correct parameter values, with:  - *Day of week for sequence* set to the current week day;  - *Mode for Sequence* field set to 0x02 (Cool mode only). | DUT SERVER unicasts a ZCL *default response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS).  If the *set weekly schedule* command frame is not accepted by the DUT SERVER, the DUT SERVER should unicasts a ZCL default response command frame to TH CLIENT with the status field equal to the appropriate value: 0x89 (INSUFFICIENT\_SPACE) if the total number of transitions  sent is greater than what the DUT SERVER supports; 0x87 (INVALID\_VALUE) if any of the set points sent in the entire sequence is out of range w.r.t. DUT SERVER’s absolute min/max setpoint limit. | |
| 6 | TSTAT.S.C02.Rsp  TSTAT.S.C00.Tx | TH CLIENT unicasts a *get weekly schedule* command frame of the *thermostat* cluster to DUT SERVER, correctly formatted with:  - *Days to return* set to the current week day;  - *Mode to return* set to 0x02 (Cool mode only). | DUT SERVER unicasts a ZCL *get weekly schedule response* command frame to TH CLIENT, carrying the data as in the *set weekly schedule* command of step 5, in addition to the data from command of step 3. | |
| 7 | TSTAT.S.C01.Rsp | TH CLIENT unicasts a *set weekly schedule* command frame of the *thermostat* cluster to DUT SERVER, correctly formatted and carrying correct parameter values, with:  - *Day of week for sequence* set to the current week day;  - *Mode for Sequence* field set to 0x03 (Heat and cool mode)*.* | DUT SERVER unicasts a ZCL *default response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS).  If the *set weekly schedule* command frame is not accepted by the DUT SERVER, the DUT SERVER should unicasts a ZCL default response command frame to TH CLIENT with the status field equal to the appropriate value: 0x89 (INSUFFICIENT\_SPACE) if the total number of transitions  sent is greater than what the DUT SERVER supports; 0x87 (INVALID\_VALUE) if any of the set points sent in the entire sequence is out of range w.r.t. DUT SERVER’s absolute min/max setpoint limit; or 0x85 (INVALID\_FIELD), if DUT SERVER cannot handle incoming command with multiple days and/or multiple modes within one command. | |
| 8 | TSTAT.S.C02.Rsp  TSTAT.S.C00.Tx | TH CLIENT unicasts a *get weekly schedule* command frame of the *thermostat* cluster to DUT SERVER, correctly formatted with:  - *Days to return* set to the current week day;  - *Mode to return* set to 0x03 (Cool mode only). | DUT SERVER unicasts a ZCL *get weekly schedule response* command frame to TH CLIENT, carrying the data as in the *set weekly schedule* command of step 7. | |
| 9 | TSTAT.S.C01.Rsp | TH CLIENT unicasts a *set weekly schedule* command frame of the *thermostat* cluster to DUT SERVER, correctly formatted and carrying correct parameter values, with:  - *Number of Transitions for Sequence* set to 2,  - *Day of week for sequence* set to the current week day;  - *Mode for Sequence* field bitmask set to 0x01 (Heat Mode),  - *Transition Time 1* set to 1 minute (0x0001), *Heat Set Point 1* set to a valid value,  - *Transition Time 2* set to 2 minutes (0x0002), *Heat Set Point 2* set to a valid value. | DUT SERVER unicasts a ZCL *default response* command frame to TH CLIENT with the *status* field equal to ?.  From the ZHA test spec: “The DUT does not accept the Set Weekly Schedule Command, verifying that the device supports 1 daily transition.”  Should it then keep the previous schedule? Test it? | |
| 10 | TSTAT.S.C01.Rsp | TH CLIENT unicasts a *set weekly schedule* command frame of the *thermostat* cluster to DUT SERVER, correctly formatted and carrying correct parameter values, with:  - *Number of Transitions for Sequence* set to 1,  - *Day of week for sequence* set to the current week day and the following day;  - *Mode for Sequence* field bitmask set to 0x01 (Heat Mode),  - *Transition Time 1* set to 1 minute (0x0001), *Heat Set Point 1* set to a valid value. | DUT SERVER unicasts a ZCL *default response* command frame to TH CLIENT with the *status* field equal to ?.  From the ZHA test spec: “The DUT does not accept the Set Weekly Schedule Command, verifying that the device supports 1 weekly transition.”  Should it then keep the previous schedule? | |
| 11 | TSTAT.S.C03.Rsp | TH CLIENT unicasts a payloadless *clear weekly schedule* command frame of the *thermostat* cluster to DUT SERVER. | DUT SERVER unicasts a ZCL *default response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS).  From the ZHA test spec “If received successfully, the DUT clears its stored weekly schedule.”  Does it need to respond with the default response? Clear weekly schedule is actually optional (according to the spec): could it fail? | |
| 12 | TSTAT.S.C04.Rsp  TSTAT.S.C01.Tx | TH CLIENT unicasts a payloadless *get relay status log* command frame of the *thermostat* cluster to DUT SERVER. | DUT SERVER unicasts a ZCL *get relay status log response* command frame to TH CLIENT, correctly formatted? | |
| *--- End of test case TSTAT-TC-03S ---* | | | |

### TSTAT-TC-04S: Thermostat cluster with separate temperature sensor and HVAC unit functionality with server as DUT

This test case verifies the primary functionality of the *thermostat* cluster server in respect to storing and updating schedules.

#### Scope

General:

* *Read attributes* command (0x00)
* *Read attributes response* command (0x01)
* *Report attributes* command (0x0a)

*Thermostat* cluster (0x0201):

* *LocalTemperature* attribute (0x0010)
* *SystemMode* attribute (0x001C)
* *PICoolingDemand* attribute (0x0007)
* *PIHeatingDemand* attribute (0x0008)
* *OccupiedCoolingSetpoint* attribute (0x0011)
* *OccupiedHeatingSetpoint* attribute (0x0012)

PICS:

* TSTAT.S
* TSTAT.S.A0007, TSTAT.S.A0008, TSTAT.S.A0010, TSTAT.S.A0011, TSTAT.S.A0012, TSTAT.S.A001C,

#### Required devices

|  |  |  |
| --- | --- | --- |
| **Designation** | **Symbol** | **Description** |
| TH1 HVAC UNIT |  | Test harness implementing:   * The *thermostat* cluster client being an HVAC unit. |
| DUT |  | Device under test server:   * The *thermostat* cluster server * also implementing *temperature measurement* cluster client |
| TH2 TEMPERATURE SENSOR |  | Test harness implementing:   * The *thermostat* cluster client being a temperature sensor. |

#### Initial conditions

|  |  |
| --- | --- |
| **Item** | **Initial Conditions** |
| 1 | A packet sniffer shall be observing the communication over the air interface. |
| 2 | All devices are factory new and powered off until used. |

#### Test preparation

|  |  |
| --- | --- |
| **Add TH1, TH2 and DUT to a ZigBee network** | |
|  |  |
| *Before* | *After* |

| **TSTAT-TC-04S: Thermostat cluster with separate temperature sensor and HVAC unit functionality with server as DUT** | | |
| --- | --- | --- |
| **Item** | **Preparation Step** | **Observation** |
| P1 | Form a ZigBee network. | Observe appropriate command frame to form the network. |
| P2 | Power on TH1, TH2 and DUT. | TH1, TH2 and DUT are powered on. |
| P3 | Join TH1, TH2 and DUT to a ZigBee network. | Observe appropriate communication between TH1, TH2, DUT and any other relevant node on the ZigBee network. |
| P4 | Establish a binding for the *thermostat* cluster on the DUT (sending Bind\_req from the TH1 HVAC UNIT or using finding&binding, as appropriate). | Observe appropriate communication between TH1 and DUT. |
| P5 | Establish a binding for the *thermostat* cluster on the TH2 TEMPERATURE SENSOR (sending Bind\_req from the TH1 or using finding&binding, as appropriate) to the DUT client. | Observe appropriate communication between TH2 and DUT. |
| P6 | TH1 unicasts a ZCL *write attributes* command frame to DUT to set the following attributes of the *thermostat* cluster:   * **0x0011** *OccupiedCoolingSetpoint* = 22°C (0x0898) * **0x0012** *OccupiedHeatingSetpoint* = 18°C (0x0708) * **0x0015** *MinHeatSetpointLimit* = 13°C (0x0514) * **0x0016** *MaxHeatSetpointLimit* = 27°C (0x0A8C) * **0x0017** *MinCoolSetpointLimit* = 13°C (0x0514) * **0x0018** *MaxCoolSetpointLimit* = 27°C (0x0A8C) * **0x001C** *SystemMode* = OFF (0x00). | DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT for each *write attributes* command, with the *status* field equal to 0x00 (SUCCESS). |
| *--- End of test case TSTAT-TC-04S preparation ---* | | |

#### Test procedure

| **TSTAT-TC-04S: Thermostat cluster with separate temperature sensor and HVAC unit functionality with server as DUT** | | | | |
| --- | --- | --- | --- | --- |
| **Item** | **PICS** | **Test Harness Step** | **DUT pass Verification** | |
| 1 | TSTAT.S.A001C | TH2 TEMPERATURE SENSOR starts unicasting to the DUT a ZCL *attribute reporting* command carrying the *LocalTemperature* attribute (**0x0000**) of the *thermostat* cluster. The *LocalTemperature* value sent shall be between the *OccupiedCoolingSetpoint* and *OccupiedHeatingSetpoint* of the DUT, as set in step P6 above.  TH1 unicasts a ZCL *read attributes* command frame to DUT to read the *SystemMode* attribute of the *thermostat* cluster. | DUT unicasts a ZCL *read attributes response* command frame to TH1 with the *status* field equal to 0x00 (SUCCESS) and the *SystemMode* attribute set to 0x00 (OFF). | |
| 2 | TSTAT.S.A0007.Report.Tx  TSTAT.S.A0008.Report.Tx | TH2 TEMPERATURE SENSOR unicasts to the DUT a ZCL *attribute reporting* command carrying the *LocalTemperature* attribute (**0x0000**) of the *thermostat* cluster, with the value above the *OccupiedCoolingSetpoint* from step P6 above. | DUT unicasts a ZCL *attribute reporting* command frame to the TH1 HVAC UNIT with the (0x0007) *PICoolingDemand* attribute with a value greater than 0% (0x64 for 100%) and (same or separate ZCL *attribute reporting* command) (0x0008) *PIHeatingDemand* attribute with a value of 0% (0x00). | |
| 3 | TSTAT.S.A001C | TH1 unicasts a ZCL *read attributes* command frame to DUT to read the *SystemMode* attribute of the *thermostat* cluster. | DUT unicasts a ZCL *read attributes response* command frame to TH1 with the *status* field equal to 0x00 (SUCCESS) and the *SystemMode* attribute set to 0x03 (Cool). | |
| 4 | TSTAT.S.A0007.Report.Tx  TSTAT.S.A0008.Report.Tx | TH2 TEMPERATURE SENSOR unicasts to the DUT a ZCL *attribute reporting* command carrying the *LocalTemperature* attribute (**0x0000**) of the *thermostat* cluster, with the value below the *OccupiedHeatingSetpoint* from step P6 above. | DUT unicasts a ZCL *attribute reporting* command frame to the TH1 HVAC UNIT with the *PIHeatingDemand* attribute (0x0008) with a value greater than 0% (0x64 for 100%) and (same or separate ZCL *attribute reporting* command) *PICoolingDemand* attribute (0x0007) with a value of 0% (0x00). | |
| 5 | TSTAT.S.A001C | TH1 unicasts a ZCL *read attributes* command frame to DUT to read the *SystemMode* attribute of the *thermostat* cluster. | DUT unicasts a ZCL *read attributes response* command frame to TH1 with the *status* field equal to 0x00 (SUCCESS) and the *SystemMode* attribute set to 0x04 (Heat). | |
| 6 | TSTAT.S.A0007.Report.Tx  TSTAT.S.A0008.Report.Tx | TH2 TEMPERATURE SENSOR starts unicasting to the DUT a ZCL *attribute reporting* command carrying the *LocalTemperature* attribute (**0x0000**) of the *thermostat* cluster. The *LocalTemperature* value sent shall be between the *OccupiedCoolingSetpoint* and *OccupiedHeatingSetpoint* of the DUT, as set in step P6 above. | DUT unicasts a ZCL *attribute reporting* command frame to the TH1 HVAC UNIT with the *PIHeatingDemand* attribute (0x0008) with a value of 0% (0x64 for 100%) and (same or separate ZCL *attribute reporting* command) *PICoolingDemand* attribute (0x0007) with a value of 0% (0x00). | |
| 7 | TSTAT.S.A001C | TH1 unicasts a ZCL *read attributes* command frame to DUT to read the *SystemMode* attribute of the *thermostat* cluster. | DUT unicasts a ZCL *read attributes response* command frame to TH1 with the *status* field equal to 0x00 (SUCCESS) and the *SystemMode* attribute set to 0x00 (OFF). | |
| 8 | TSTAT.S.A0011 | Via UI on the DUT, the DUT’s *OccupiedCoolingSetpoint* is set to 13°C (0x0514).  TH1 unicasts a ZCL *read attributes* command frame to DUT to read the *OccupiedCoolingSetpoint* attribute of the DUT. | DUT unicasts a ZCL *read attributes response* command frame to TH1 with the *status* field equal to 0x00 (SUCCESS) and the *OccupiedCoolingSetpoint* attribute set to 13°C (0x0514). | |
| 9 | TSTAT.S.A0011 | Via UI on the DUT, the DUT’s *OccupiedCoolingSetpoint* is set to 12°C (0x04B0).  TH1 unicasts a ZCL *read attributes* command frame to DUT to read the *OccupiedCoolingSetpoint* attribute of the DUT. | DUT unicasts a ZCL *read attributes response* command frame to TH1 with the *status* field equal to 0x00 (SUCCESS) and the *OccupiedCoolingSetpoint* attribute in NOT set to 12°C (0x04B0). | |
| 10 | TSTAT.S.A0011 | Via UI on the DUT, the DUT’s *OccupiedCoolingSetpoint* is set to 27°C (0x0A8C).  TH1 unicasts a ZCL *read attributes* command frame to DUT to read the *OccupiedCoolingSetpoint* attribute of the DUT. | DUT unicasts a ZCL *read attributes response* command frame to TH1 with the *status* field equal to 0x00 (SUCCESS) and the *OccupiedCoolingSetpoint* attribute set to 27°C (0x0A8C). | |
| 11 | TSTAT.S.A0011 | Via UI on the DUT, the DUT’s *OccupiedCoolingSetpoint* is set to 28°C (0x0AF0).  TH1 unicasts a ZCL *read attributes* command frame to DUT to read the *OccupiedCoolingSetpoint* attribute of the DUT. | DUT unicasts a ZCL *read attributes response* command frame to TH1 with the *status* field equal to 0x00 (SUCCESS) and the *OccupiedCoolingSetpoint* attribute is NOT set to 28°C (0x0AF0). | |
| 12 | TSTAT.S.A0012 | Via UI on the DUT, the DUT’s *OccupiedHeatingSetpoint* is set to 13°C (0x0514).  TH1 unicasts a ZCL *read attributes* command frame to DUT to read the *OccupiedHeatingSetpoint* attribute of the DUT. | DUT unicasts a ZCL *read attributes response* command frame to TH1 with the *status* field equal to 0x00 (SUCCESS) and the *OccupiedHeatingSetpoint* attribute is set to 13°C (0x0514). | |
| 13 | TSTAT.S.A0012 | Via UI on the DUT, the DUT’s *OccupiedHeatingSetpoint* is set to 12°C (0x04B0).  TH1 unicasts a ZCL *read attributes* command frame to DUT to read the *OccupiedHeatingSetpoint* attribute of the DUT. | DUT unicasts a ZCL *read attributes response* command frame to TH1 with the *status* field equal to 0x00 (SUCCESS) and the *OccupiedHeatingSetpoint* attribute is NOT set to 12°C (0x04B0). | |
| 14 | TSTAT.S.A0012 | Via UI on the DUT, the DUT’s *OccupiedHeatingSetpoint* is set to 27°C (0x0A8C).  TH1 unicasts a ZCL *read attributes* command frame to DUT to read the *OccupiedHeatingSetpoint* attribute of the DUT. | DUT unicasts a ZCL *read attributes response* command frame to TH1 with the *status* field equal to 0x00 (SUCCESS) and the *OccupiedHeatingSetpoint* attribute is set to 27°C (0x0A8C) | |
| 15 | TSTAT.S.A0012 | Via UI on the DUT, the DUT’s *OccupiedHeatingSetpoint* is set to 28°C (0x0AF0).  TH1 unicasts a ZCL *read attributes* command frame to DUT to read the *OccupiedHeatingSetpoint* attribute of the DUT. | DUT unicasts a ZCL *read attributes response* command frame to TH1 with the *status* field equal to 0x00 (SUCCESS) and the *OccupiedHeatingSetpoint* attribute is NOT set to 28°C (0x0AF0). | |
| *--- End of test case TSTAT-TC-04S ---* | | | |

### TSTAT-TC-05S: Scenes functionality with server as DUT (!!!)

This test case verifies the scenes functionality of the *thermostat* cluster server.

#### Scope

General:

* *Read attributes* command (0x00)
* *Read attributes response* command (0x01)
* *Default response* command (0x0b)

*Groups* cluster (0x0004):

* *Add group* command (0x00)
* *Add group response* command (0x00)
* *Get group membership* command (0x02)
* *Get group membership response* command (0x02)
* *Remove all groups* command (0x04)

*Scenes* cluster (0x0005):

* *Remove all scenes* command (0x03)
* *Remove all scenes response* command (0x03)
* *Store scene* command (0x04)
* *Store scene response* command (0x04)
* *Recall scene* command (0x05)

*Thermostat* cluster (0x0201):

* *OccupiedCoolingSetpoint* attribute (0x0011)
* *OccupiedHeatingSetpoint* attribute (0x0012)
* *SystemMode* attribute (0x001C)

PICS:

* G.S, S.S, OO.S
* G.S.C00.Rsp, G.S.C02.Rsp-G.S.C04.Rsp
* G.S.C00.Tx, G.S.C02.Tx, G.S.C03.Tx
* S.S.C04.Rsp, S.S.C05.Rsp
* S.S.C04.Tx
* TSTAT.S.A0011, TSTAT.S.A0011.Scene, TSTAT.S.A0012, TSTAT.S.A0012.Scene, TSTAT.S.A001C, TSTAT.S.A001C.Scene

#### Required devices

|  |  |  |
| --- | --- | --- |
| **Designation** | **Symbol** | **Description** |
| TH CLIENT |  | Test harness client implementing:   * The *groups* cluster client, * The *scenes* cluster client and * The *thermostat* cluster client. |
| DUT SERVER |  | Device under test server implementing:   * The *groups* cluster server, * The *scenes* cluster server and * The *thermostat* cluster server. |

#### Initial conditions

|  |  |
| --- | --- |
| **Item** | **Initial Conditions** |
| 1 | A packet sniffer shall be observing the communication over the air interface. |
| 2 | All devices are factory new and powered off until used. |

#### Test preparation

|  |  |
| --- | --- |
| **Link TH CLIENT with DUT SERVER** | |
|  |  |
| *Before* | *After* |

| **TSTAT-TC-05S: Scenes functionality with server as DUT** | | |
| --- | --- | --- |
| **Item** | **Preparation Step** | **Observation** |
| P1 | Form a ZigBee network. | Observe appropriate command frame to form the network. |
| P2 | Power on TH CLIENT and DUT SERVER. | TH CLIENT and DUT SERVER are powered on. |
| P3 | Join TH CLIENT and DUT SERVER to a ZigBee network. | Observe appropriate communication between TH CLIENT, DUT SERVER and any other relevant node on the ZigBee network. |
| *--- End of test case TSTAT-TC-05S preparation ---* | | |

#### Test procedure

| **TSTAT-TC-05S: Scene functionality with server as DUT** | | | | |
| --- | --- | --- | --- | --- |
| **Item** | **PICS** | **Test Harness Step** | **DUT Pass Verification** | |
| 1a | G.S.C04.Rsp | TH CLIENT unicasts a ZCL *remove all groups* command frame to DUT SERVER. | If requested, DUT SERVER unicasts a ZCL *default response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS). | |
| 1b | G.S.C02.Rsp, G.S.C02.Tx | TH CLIENT unicasts a ZCL *get group membership* command frame to DUT SERVER with the *group count* field set to 0x00. | DUT SERVER unicasts a ZCL *get group membership response* command frame with the *group count* field equal to 0x00. | |
| 1c | G.S.C00.Rsp, G.S.C00.Tx | TH CLIENT unicasts ZCL *add group* command to DUT SERVER, with the *group ID* field set to 0x0001. | DUT SERVER unicasts a ZCL *add group response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS) and the *group ID* field equal to 0x0001. | |
| 2 | S.S.C03.Rsp, S.S.C03.Tx | TH CLIENT unicasts a ZCL *remove all scenes* command frame to DUT SERVER with the *group ID* field set to 0x0001. | DUT SERVER unicasts a ZCL *remove all scenes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS) and the *group ID* field equal to 0x0001. | |
| 3a | TSTAT.S.A0011,  TSTAT.S.A0012,  TSTAT.S.A001C | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER, to set:  - *OccupiedHeatingSetpoint* attribute to 0x06a4 (17°C);  - *OccupiedCoolingSetpoint* attribute to 0x07d0 (20°C);  *- SystemMode* attribute to 0x01 (Auto). | If requested, DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS).  DUT SERVER starts adapting the temperature. | |
| 3b | TSTAT.S.A0011.Scene,  TSTAT.S.A0012.Scene,  TSTAT.S.A001C.Scene, S.S.C04.Rsp, S.S.C04.Tx | TH CLIENT unicasts a ZCL *store scene* command frame to DUT SERVER with the *group ID* field set to 0x0001 and the *scene ID* field set to 0x01. | DUT SERVER unicasts a ZCL *store scene response* command frame to TH CLIENT with the *status* field set to 0x00 (SUCCESS), the *group ID* field set to 0x0001 and the *scene ID* field set to 0x01. | |
| 3c | TSTAT.S.A0011,  TSTAT.S.A0012.  TSTAT.S.A001C | TH CLIENT unicasts a ZCL *read attributes* command frame for the *OccupiedCoolingSetpoint, OccupiedHeatinSetpoint* and *SystemMode* attribute to DUT SERVER. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT.  The attributes have the values as written in step 3b. | |
| 3d | TSTAT.S.A0011,  TSTAT.S.A0012.  TSTAT.S.A001C | TH CLIENT unicasts a ZCL *write attributes* command frame to DUT SERVER, to set:  - *OccupiedHeatingSetpoint* attribute to 0x07d0 (20°C);  - *OccupiedCoolingSetpoint* attribute to 0x08fc (23°C);  *- SystemMode* attribute to 0x00 (Off). | If requested, DUT SERVER unicasts a ZCL *write attributes response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS).  DUT SERVER turns off. | |
| 3e | TSTAT.S.A0011,  TSTAT.S.A0012.  TSTAT.S.A001C | TH CLIENT unicasts a ZCL *read attributes* command frame for the *OccupiedCoolingSetpoint, OccupiedHeatinSetpoint* and *SystemMode* attribute to DUT SERVER. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT.  The attributes have the values as written in step 3d. | |
| 3f | TSTAT.S.A0011.Scene,  TSTAT.S.A0012.Scene,  TSTAT.S.A001C.Scene, S.S.C05.Rsp | TH CLIENT unicasts a ZCL *recall scene* command frame to DUT SERVER with the *group ID* field set to 0x0001 and the *scene ID* field set to 0x01. | If requested, DUT SERVER unicasts a ZCL *default response* command frame to TH CLIENT with the *status* field equal to 0x00 (SUCCESS).  DUT SERVER starts adapting the temperature. | |
| 3g | TSTAT.S.A0011,  TSTAT.S.A0012.  TSTAT.S.A001C | TH CLIENT unicasts a ZCL *read attributes* command frame for the *OccupiedCoolingSetpoint, OccupiedHeatinSetpoint* and *SystemMode* attribute to DUT SERVER. | DUT SERVER unicasts a ZCL *read attributes response* command frame to TH CLIENT.  The attributes have the values as written in step 3b. | |
| *--- End of test case TSTAT-TC-05S ---* | | | |

### TSTAT-TC-06S: Reporting functionality with server as DUT

This case test verifies the attribute reporting behavior of the *thermostat* cluster server.

Test 5.20.5 of ZHA test specification (07-5340r15) not copied, pending clarification.

This test case verifies the primary functionality of the *thermostat* cluster server in respect to storing and updating schedules.

#### Scope

General:

* *Configure reporting* command (0x06)
* *Configure reporting response* command (0x07)
* *Read reporting configuration* command (0x08)
* *Read reporting configuration response* command (0x09)
* *Report attributes* command (0x0a)

*Thermostat* cluster (0x0201):

* *LocalTemperature* attribute (0x0000)
* *PICoolingDemand* attribute (0x0007)
* *PICoolingDemand* attribute (0x0008)

PICS:

* TSTAT.S
* TSTAT.S.A0000, TSTAT.S.A0000.Report.Tx, TSTAT.S.A0007, TSTAT.S.A0007.Report.Tx, TSTAT.S.A0008, TSTAT.S.A0008.Report.Tx,

#### Required devices

|  |  |  |
| --- | --- | --- |
| **Designation** | **Symbol** | **Description** |
| TH CLIENT |  | Test harness client implementing:   * The *thermostat* cluster client. |
| DUT SERVER |  | Device under test server:   * The *thermostat* cluster server. |
| - | - | Heat and cooling source (warm finger and coolant/freeze-spray can do the trick) |

#### Initial conditions

|  |  |
| --- | --- |
| **Item** | **Initial Conditions** |
| 1 | A packet sniffer shall be observing the communication over the air interface. |
| 2 | All devices are factory new and powered off until used. |

#### Test preparation

|  |  |
| --- | --- |
| **Add TH CLIENT and DUT SERVER to a ZigBee network** | |
|  |  |
| *Before* | *After* |

| **TSTAT-TC-06S: Reporting attributes with server as a DUT** | | |
| --- | --- | --- |
| **Item** | **Preparation Step** | **Observation** |
| P1 | Form a ZigBee network. | Observe appropriate command frame to form the network. |
| P2 | Power on TH CLIENT and DUT SERVER. | TH CLIENT and DUT SERVER are powered on. |
| P3 | Join TH CLIENT and DUT SERVER to a ZigBee network. | Observe appropriate communication between TH CLIENT, DUT SERVER and any other relevant node on the ZigBee network. |
| P4 | Establish a binding link from an endpoint on DUT SERVER to a corresponding endpoint on TH CLIENT that both support the *thermostat* cluster. | Observe appropriate communication between DUT SERVER, TH CLIENT and any other relevant node on the ZigBee network. |
| *--- End of test case TSTAT-TC-06S preparation ---* | | |

#### Test procedure

| **TSTAT-TC-06S: Reporting attributes with server as a DUT** | | | | |
| --- | --- | --- | --- | --- |
| **Item** | **PICS** | **Test Harness Step** | **DUT pass Verification** | |
| *LocalTemperature* attribute | | | | |
| 1a | TSTAT.S.A0000  TSTAT.S.A0000.Report.Tx | TH CLIENT unicasts a ZCL *read reporting configuration* command to DUT SERVER for the *LocalTemperature* attribute of the *thermostat* cluster. | DUT SERVER unicasts a ZCL *read* *reporting configuration response* command to TH CLIENT, carrying default reporting configuration for the *LocalTemperature* attribute of the *thermostat* cluster, with:  - *Status* field se to 0x00 (SUCCESS);  *- Direction* field set to 0x00 (reported attribute);  - *Attribute Data Type* field present and set to 0x29;  - *Minimum* and *Maximum* *reporting interval* fields present;  - *Reportable change* field present;  - *Timeout period* field omitted. | |
| 1b | TSTAT.S.A0000  TSTAT.S.A0000.Report.Tx | If practical (depending on the reporting interval of the default reporting configuration), wait for the attribute report according to default configuration. | At a time as specified by the default reporting configuration of step 0a, DUT SERVER unicasts a ZCL *report attributes* command to TH CLIENT with the *LocalTemperature* attribute. | |
| 2a | TSTAT.S.A0000  TSTAT.S.A0000.Report.Tx | TH CLIENT unicasts a ZCL *configure reporting* command to DUT SERVER for the *LocalTemperature* attribute with a *direction* field set to 0x00, the *minimum reporting interval* field set to TSTAT.PIXIT.LOCALTEMPERATURE.MINRI (suggested 2 seconds), the *maximum reporting interval* field set to TSTAT.PIXIT.LOCALTEMPERATURE.MAXRI (suggested 5 seconds) and the *reportable change* field set to 0x0001. | DUT SERVER unicasts a ZCL *configure* *reporting response* command to TH CLIENT, confirming the configured attribute and with the *status* field set to SUCCESS. | |
| 2b | TSTAT.S.A0000  TSTAT.S.A0000.Report.Tx | Keep the temperature of the DUT SERVER constant. | Approx. every TSTAT.PIXIT.LOCALTEMPERATURE.MAXRI seconds, DUT SERVER unicasts a ZCL *report attributes* command to TH CLIENT with the *LocalTemperature* attribute. | |
| 2c | TSTAT.S.A0000  TSTAT.S.A0000.Report.Tx | Via heat source, increase the temperature of the DUT SERVER. | Approx. every TSTAT.PIXIT.LOCALTEMPERATURE.MINRI seconds (not faster), DUT SERVER unicasts a ZCL *report attributes* command to TH CLIENT with the *LocalTemperature* attribute. | |
| *PICoolingDemand* attribute | | | | |
| 3a | TSTAT.S.A0007  TSTAT.S.A0007.Report.Tx | TH CLIENT unicasts a ZCL *read reporting configuration* command to DUT SERVER for the *PICoolingDemand* attribute of the *thermostat* cluster. | DUT SERVER unicasts a ZCL *read* *reporting configuration response* command to TH CLIENT, carrying default reporting configuration for the *PICoolingDemand* attribute of the *thermostat* cluster, with:  - *Status* field se to 0x00 (SUCCESS);  *- Direction* field set to 0x00 (reported attribute);  - *Attribute Data Type* field present and set to 0x20;  - *Minimum* and *Maximum* *reporting interval* fields present;  - *Reportable change* field present;  - *Timeout period* field omitted. | |
| 3b | TSTAT.S.A0007 TSTAT.S.A0007.Report.Tx | If practical (depending on the reporting interval of the default reporting configuration), wait for the attribute report according to default configuration. | At a time as specified by the default reporting configuration of step 0a, DUT SERVER unicasts a ZCL *report attributes* command to TH CLIENT with the *PICoolingDemand* attribute. | |
| 4a | TSTAT.S.A0007 TSTAT.S.A0007.Report.Tx | TH CLIENT unicasts a ZCL *configure reporting* command to DUT SERVER for the *PICoolingDemand* attribute with a *direction* field set to 0x00, the *minimum reporting interval* field set to TSTAT.PIXIT.PICOOLINGDEMAND.MINRI(suggested 2 seconds), the *maximum reporting interval* field set to TSTAT.PIXIT.PICOOLINGDEMAND.MAXRI (suggested 5 seconds) and the *reportable change* field set to 0x01. | DUT SERVER unicasts a ZCL *configure* *reporting response* command to TH CLIENT, confirming the configured attribute and with the *status* field set to SUCCESS. | |
| 4b | TSTAT.S.A0007 TSTAT.S.A0007.Report.Tx | Using method described in the OEM supplied PIXIT, hold the *PICoolingDemand* at a constant value. | Approx. every TSTAT.PIXIT.PICOOLINGDEMAND.MAXRI seconds, DUT SERVER unicasts a ZCL *report attributes* command to TH CLIENT with the *PICoolingDemand* attribute. | |
| 4c | TSTAT.S.A0007 TSTAT.S.A0007.Report.Tx | Using method described in the OEM supplied PIXIT, set the thermostat into a state such that the *PICoolingDemand* attribute is rapidly changing. | Approx. every TSTAT.PIXIT.PICOOLINGDEMAND.MINRI seconds (not faster), DUT SERVER unicasts a ZCL *report attributes* command to TH CLIENT with the *PICoolingDemand* attribute. | |
| *PIHeatingDemand* attribute | | | | |
| 5a | TSTAT.S.A0008  TSTAT.S.A0008.Report.Tx | TH CLIENT unicasts a ZCL *read reporting configuration* command to DUT SERVER for the *PIHeatingDemand* attribute of the *thermostat* cluster. | DUT SERVER unicasts a ZCL *read* *reporting configuration response* command to TH CLIENT, carrying default reporting configuration for the *PIHeatingDemand* attribute of the *thermostat* cluster, with:  - *Status* field se to 0x00 (SUCCESS);  *- Direction* field set to 0x00 (reported attribute);  - *Attribute Data Type* field present and set to 0x20;  - *Minimum* and *Maximum* *reporting interval* fields present;  - *Reportable change* field present;  - *Timeout period* field omitted. | |
| 5b | TSTAT.S.A0008  TSTAT.S.A0008.Report.Tx | If practical (depending on the reporting interval of the default reporting configuration), wait for the attribute report according to default configuration. | At a time as specified by the default reporting configuration of step 0a, DUT SERVER unicasts a ZCL *report attributes* command to TH CLIENT with the *PIHeatingDemand* attribute. | |
| 6a | TSTAT.S.A0008  TSTAT.S.A0008.Report.Tx | TH CLIENT unicasts a ZCL *configure reporting* command to DUT SERVER for the *PIHeatingDemand* attribute with a *direction* field set to 0x00, the *minimum reporting interval* field set to TSTAT.PIXIT.PIHEATINGDEMAND.MINRI (suggested 2 seconds), the *maximum reporting interval* field set to TSTAT.PIXIT.PIHEATINGDEMAND.MAXRI (suggested 5 seconds) and the *reportable change* field set to 0x01. | DUT SERVER unicasts a ZCL *configure* *reporting response* command to TH CLIENT, confirming the configured attribute and with the *status* field set to SUCCESS. | |
| 6b | TSTAT.S.A0008  TSTAT.S.A0008.Report.Tx | Using method described in the OEM supplied PIXIT, hold the *PIHeatingDemand* at a constant value. | Approx. every TSTAT.PIXIT.PIHEATINGDEMAND.MAXRI seconds, DUT SERVER unicasts a ZCL *report attributes* command to TH CLIENT with the *PIHeatingDemand* attribute. | |
| 6c | TSTAT.S.A0008  TSTAT.S.A0008.Report.Tx | Using method described in the OEM supplied PIXIT, set the thermostat into a state such that the *PIHeatingDemand* attribute is rapidly changing. | Approx. every TSTAT.PIXIT.PIHEATINGDEMAND.MINRI seconds (not faster), DUT SERVER unicasts a ZCL *report attributes* command to TH CLIENT with the *PIHeatingDemand* attribute. | |
| *--- End of test case TSTAT-TC-06S ---* | | | |

## Client test cases

### TSTAT-TC-01C: Functionality with client as DUT (TBD)

This case test verifies the functionality of the *thermostat* cluster client.

The DUT client SHALL be on the same network as a suitable server, provided by the user, and this device SHALL be used by the client to exercise its functionality. The test case uses the test harness to prompt the user, based on the declared PICS, to exercise the functionality of the *thermostat* cluster client and to verify the results. A sniffer tool SHALL be used to log the exercised functionality and to determine its validity.

In this test case, the PICS notation TSTAT.C.C*d*,Txrepresents the list of commands that are declared as being transmitted by the DUT.

#### Scope

PICS:

* TSTAT.C
* TSTAT.C.A0000.Report.Rsp
* TSTAT.C.A0003.Report.Rsp

#### Required devices

|  |  |  |
| --- | --- | --- |
| **Designation** | **Symbol** | **Description** |
| DUT CLIENT |  | Device under test client implementing:   * The *thermostat* cluster client. |
| SERVER |  | Suitable server device implementing:   * The *thermostat* cluster server. |

#### Initial conditions

|  |  |
| --- | --- |
| **Item** | **Initial Conditions** |
| 1 | A packet sniffer shall be observing the communication over the air interface. |
| 2 | All devices are factory new and powered off until used. |

#### Test preparation

|  |  |
| --- | --- |
| **Link DUT CLIENT with SERVER** | |
|  |  |
| *Before* | *After* |

| **TSTAT-TC-01C: Functionality with client as DUT** | | |
| --- | --- | --- |
| **Item** | **Preparation Step** | **Observation** |
| P1 | Power on the DUT CLIENT device and the SERVER device. | DUT CLIENT and SERVER are powered on. |
| P2 | Ensure the DUT CLIENT device and the SERVER device are on the same ZigBee network. | Observe appropriate communication between DUT CLIENT, SERVER and any other relevant node on the ZigBee network. |
| *--- End of test case TSTAT-TC-01C preparation ---* | | |

#### Test procedure

| **TSTAT-TC-01C: Functionality with client as DUT** | | | |
| --- | --- | --- | --- |
| **Item** | **PICS** | **Test Harness Step** | **DUT Pass Verification** |
| 1 | - | Test harness prompts the user with a list of commands, based on the declared PICS, which the DUT CLIENT indicates it can transmit. | None. |
| 2 | TSTAT.C.C*d*.Tx | None. | DUT CLIENT transmits correctly formed commands in any order and with application achievable values. This is verified using the sniffer log. |
| 3 | - | Prompt the user to verify that the cluster commands listed in step 1 were transmitted during step 2. | During step 2, DUT CLIENT has transmitted every command listed by the test harness in step 1. |
| 4 | - | Prompt the user to verify that the cluster commands not listed in step 1 were not transmitted during step 2. | During step 2, DUT CLIENT has not transmitted any commands from this cluster that were not listed by the test harness in step 1. |
| *--- End of test case TSTAT-TC-01C ---* | | | |

# Annex A: PICS to test case cross reference (TBD)

## Server

|  | **Test case** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **PICS** | TSTAT-TC-01G | TSTAT-TC-01S | TSTAT-TC-02S | TSTAT-TC-03S | TSTAT-TC-04S | TSTAT-TC-05S | TSTAT-TC-06S |
| TSTAT.S | X | X | X | X | X | X | X |
| TSTAT.S.A0000 |  | X | X |  |  |  | X |
| TSTAT.S.A0000.Report.Tx |  |  |  |  |  |  | X |
| TSTAT.S.A0001 |  | X |  |  |  |  |  |
| TSTAT.S.A0002 |  | X |  |  |  |  |  |
| TSTAT.S.A0003 |  | X | X |  |  |  |  |
| TSTAT.S.A0004 |  | X | X |  |  |  |  |
| TSTAT.S.A0005 |  | X | X |  |  |  |  |
| TSTAT.S.A0006 |  | X | X |  |  |  |  |
| TSTAT.S.A0007 |  | X |  |  | X |  | X |
| TSTAT.S.A0007.Report.Tx |  |  |  |  |  |  | X |
| TSTAT.S.A0008 |  | X |  |  | X |  | X |
| TSTAT.S.A0008.Report.Tx |  |  |  |  |  |  | X |
| TSTAT.S.A0009 |  | X | X |  |  |  |  |
| TSTAT.S.A0010 |  | X | X |  | X |  |  |
| TSTAT.S.A0011 |  | X | X |  | X | X |  |
| TSTAT.S.A0011.Scene |  |  |  |  |  | X |  |
| TSTAT.S.A0012 |  | X | X |  | X | X |  |
| TSTAT.S.A0012.Scene |  |  |  |  |  | X |  |
| TSTAT.S.A0013 |  | X | X |  |  |  |  |
| TSTAT.S.A0014 |  | X | X |  |  |  |  |
| TSTAT.S.A0015 |  | X | X |  |  |  |  |
| TSTAT.S.A0016 |  | X | X |  |  |  |  |
| TSTAT.S.A0017 |  | X | X |  |  |  |  |
| TSTAT.S.A0018 |  | X | X |  |  |  |  |
| TSTAT.S.A0019 |  | X | X |  |  |  |  |
| TSTAT.S.A001A |  | X | X |  |  |  |  |
| TSTAT.S.A001B |  | X | X |  |  |  |  |
| TSTAT.S.A001C |  | X | X |  | X | X |  |
| TSTAT.S.A001C.Scene |  |  |  |  |  | X |  |
| TSTAT.S.A001D |  | X |  |  |  |  |  |
| TSTAT.S.A001E |  | X |  |  |  |  |  |
| TSTAT.S.A0020 |  | X |  |  |  |  |  |
| TSTAT.S.A0021 |  | X |  |  |  |  |  |
| TSTAT.S.A0022 |  | X |  |  |  |  |  |
| TSTAT.S.A0023 |  | X |  | X |  |  |  |
| TSTAT.S.A0024 |  | X |  | X |  |  |  |
| TSTAT.S.A0025 |  | X |  |  |  |  |  |
| TSTAT.S.A0029 |  | X |  |  |  |  |  |
| TSTAT.S.A0030 |  | X |  |  |  |  |  |
| TSTAT.S.A0031 |  | X |  |  |  |  |  |
| TSTAT.S.A0032 |  | X |  |  |  |  |  |
| TSTAT.S.A0040 |  | X |  |  |  |  |  |
| TSTAT.S.A0041 |  | X |  |  |  |  |  |
| TSTAT.S.A0042 |  | X |  |  |  |  |  |
| TSTAT.S.A0043 |  | X |  |  |  |  |  |
| TSTAT.S.A0044 |  | X |  |  |  |  |  |
| TSTAT.S.A0045 |  | X |  |  |  |  |  |
| TSTAT.S.A0046 |  | X |  |  |  |  |  |
| TSTAT.S.A0047 |  | X |  |  |  |  |  |
| TSTAT.S.Afffd | X | X |  |  |  |  |  |
| TSTAT.S.Afffe | X | X |  |  |  |  |  |
| TSTAT.S.C00.Rsp |  |  | X |  |  |  |  |
| TSTAT.S.C01.Rsp |  |  |  | X |  |  |  |
| TSTAT.S.C02.Rsp |  |  |  | X |  |  |  |
| TSTAT.S.C03.Rsp |  |  |  | X |  |  |  |
| TSTAT.S.C04.Rsp |  |  |  | X |  |  |  |
| TSTAT.S.C00.Tx |  |  |  | X |  |  |  |
| TSTAT.S.C01.Tx |  |  |  | X |  |  |  |

## Client

|  | **Test case** | |
| --- | --- | --- |
| **PICS** | TSTAT-TC-01G | TSTAT-TC-01C |
| TSTAT.C | X | X |
| TSTAT.S.A0000.Report.Rsp |  | X |
| TSTAT.S.A0003.Report.Rsp |  | X |
| TSTAT.C.Afffd | X |  |
| TSTAT.C.Afffe | X |  |

1. M.1: the DUT SHALL implement at least one of the M.1 attributes. [↑](#footnote-ref-2)