AllJoyn™ Service Framework 16.04 Test Case Specification

March 10, 2016

Contents

[1 Introduction 4](#_Toc451437650)

[1.1 Purpose 4](#_Toc451437651)

[1.2 Scope 4](#_Toc451437652)

[1.3 References 4](#_Toc451437653)

[2 Environment setup 5](#_Toc451437654)

[2.1 Requirements 5](#_Toc451437655)

[2.2 Preconditions 5](#_Toc451437656)

[2.3 Parameters 5](#_Toc451437657)

[3 HAE service framework test cases 6](#_Toc451437658)

[3.1 HAE-v1-01: HAEAboutData Test 6](#_Toc451437659)

[3.2 HAE-v1-02: AirRecirculationMode Interface Test 8](#_Toc451437660)

[3.3 HAE-v1-03: AudioVideoInput Interface Test 9](#_Toc451437661)

[3.4 HAE-v1-04: AudioVolume Interface Test 10](#_Toc451437662)

[3.5 HAE-v1-05: BatteryStatus Interface Test 12](#_Toc451437663)

[3.6 HAE-v1-06: Channel Interface Test 13](#_Toc451437664)

[3.7 HAE-v1-07: ClimateControlMode Interface Test 15](#_Toc451437665)

[3.8 HAE-v1-08: ClosedStatus Interface Test 17](#_Toc451437666)

[3.9 HAE-v1-09: CurrentPower Interface Test 18](#_Toc451437667)

[3.10 HAE-v1-10: CurrentTemperature Interface Test 19](#_Toc451437668)

[3.11 HAE-v1-11: CycleControl Interface Test 20](#_Toc451437669)

[3.12 HAE-v1-12: DishWashingCyclePhase Interface Test 21](#_Toc451437670)

[3.13 HAE-v1-13: EnergyUsage Interface Test 22](#_Toc451437671)

[3.14 HAE-v1-14: FanSpeedLevel Interface Test 23](#_Toc451437672)

[3.15 HAE-v1-15: HeatingZone Interface Test 25](#_Toc451437673)

[3.16 HAE-v1-16: Hid Interface Test 26](#_Toc451437674)

[3.17 HAE-v1-17: LaundryCyclePhase Interface Test 27](#_Toc451437675)

[3.18 HAE-v1-18: OffControl Interface Test 28](#_Toc451437676)

[3.19 HAE-v1-19: OnControl Interface Test 29](#_Toc451437677)

[3.20 HAE-v1-20: OnOffStatus Interface Test 30](#_Toc451437678)

[3.21 HAE-v1-21: OvenCyclePhase Interface Test 31](#_Toc451437679)

[3.22 HAE-v1-22: RapidMode Interface Test 32](#_Toc451437680)

[3.23 HAE-v1-23: RemoteControllability Interface Test 33](#_Toc451437681)

[3.24 HAE-v1-24: RepeatMode Interface Test 34](#_Toc451437682)

[3.25 HAE-v1-25: ResouceSaving Interface Test 35](#_Toc451437683)

[3.26 HAE-v1-26: RobotCleaningCyclePhase Interface Test 36](#_Toc451437684)

[3.27 HAE-v1-27: SoilLevel Interface Test 37](#_Toc451437685)

[3.28 HAE-v1-28: SpinSpeedLevel Interface Test 39](#_Toc451437686)

[3.29 HAE-v1-29: TargetTemperature Interface Test 41](#_Toc451437687)

[3.30 HAE-v1-30: Timer Interface Test 43](#_Toc451437688)

[3.31 HAE-v1-31: WaterLevel Interface Test 44](#_Toc451437689)

[3.32 HAE-v1-32: WindDirection Interface Test 45](#_Toc451437690)

[Appendix A. Implementation Conformance Statement, ICS 48](#_Toc451437691)

[Appendix B. Implementation eXtra Information for Testing, IXIT 51](#_Toc451437692)

[Appendix C. Test Case Mapping Table, TCMT 53](#_Toc451437693)

# Introduction

## Purpose

These test cases evaluate and verify the functionality related to the AllJoyn™ Home Appliances and Entertainment (HAE) service framework 16.04 collection of interfaces.

The About interface is required by an application to provide the discovery mechanism for the service framework interfaces that it supports, as well as providing the basic identification information.

## Scope

These test cases are designed to determine if a device conforms to the HAE service framework 16.04 interface specifications. Successful completion of all test cases in this document does not guarantee that the tested device will interoperate with other devices.

## References

The following are reference documents.

AllJoyn™ HAE Service Framework 16.04 Interface Specification

# Environment setup

## Requirements

The following are required in order to execute these test cases:

An AllJoyn-enabled device (the device under test or DUT) that implements the HAE service framework 16.04 and supports the About feature 1.0

* A supported test device on which the test cases will run
* A Wi-Fi access point (referred to as the personal AP)

## Preconditions

Before running these test cases, it is assumed that:

* The DUT is connected to the personal AP
* The test device is connected to the personal AP
* At least one application on the DUT is announcing its capabilities through an About announcement

## Parameters

Table . Parameters for the HAE service framework

| Parameter | Description |
| --- | --- |
| DeviceId | Device ID of the DUT |
| AppId | Application ID of the application on the DUT |

# HAE service framework test cases

## HAE-v1-01: HAEAboutData Test

Objective

Verify Hae AboutData of the DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. Check additional About metadata fields.
   * Check existence of mandatory fields as followings :
     + Location field
     + DeviceTypeDescription field
3. Check existence of minimum required interfaces of device type.

Expected results

* The test device receives an About announcement from the application on the DUT.
* There are Location field and DeviceTypeDescription field in About data.
* There are minimum required interfaces of DeviceType on the DUT as Table 2

Table . Minimum required interfaces (ref. theory-of-operation-v1.md)

| Device Type | Id | Minimum required interfaces |
| --- | --- | --- |
| Root | 0 | None |
| Other | 1 | None |
| Refrigerator | 2 | Operation.ClosedStatus |
| Freezer | 3 | Operation.ClosedStatus |
| Ice Maker | 4 | None |
| Air Conditioner | 5 | Operation.OnOffStatus, Operation.ClimateControlMode, Environment.TargetTemperature, Environment.CurrentTemperature |
| Thermostat | 6 | Operation.OnOffStatus, Operation.ClimateControlMode, Environment.TargetTemperature, Environment.CurrentTemperature |
| Humidifier | 7 | Operation.OnOffStatus |
| Dehumidifier | 8 | Operation.OnOffStatus, Environment.TargetHumidity, Environment.CurrentHumidity |
| Air Purifier | 9 | Operation.OnOffStatus |
| Electric Fan | 10 | Operation.OnOffStatus |
| Air Quality Monitor | 11 | At least one between Environment.CurrentAirQuality and Environment.CurrentAirQualityLevel |
| Clothes Washer | 12 | Operation.OnOffStatus, Operation.CycleControl |
| Clothes Dryer | 13 | Operation.OnOffStatus, Operation.CycleControl |
| Clothes Washer-Dryer | 14 | Operation.OnOffStatus, Operation.CycleControl |
| Dish Washer | 15 | Operation.OnOffStatus, Operation.CycleControl |
| Robot Cleaner | 16 | Operation.OnOffStatus, Operation.CycleControl |
| Oven | 17 | Operation.OnOffStatus, Operation.CycleControl |
| Cooker Hood | 18 | Operation.OnOffStatus, Operation.HvacMode, Operation.FanSpeedLevel |
| Cooktop | 19 | Operation.HeatingZone |
| FoodProbe | 20 | Environment.TargetTemperature |
| Television | 21 | Operation.OnOffStatus, Operation.Channel, Operation.AudioVolume |
| SetTopBox | 22 | Operation.OnOffStatus |

## HAE-v1-02: AirRecirculationMode Interface Test

Objective

Verify the properties, methods and signals of the AirRecirculationModeinterface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is AirRecirculationMode Interface on DUT.
3. Get initial values for all properties.
   * Retrieve the IsRecirculating property.
4. Initialize all read-write properties.
   * Initializethe IsRecirculating property to true.
5. Set properties to valid value.
   * Set the IsRecirculating property to false.
     + Wait the PropertiesChanged signal for the IsRecirculating property.
     + Get the IsRecirculating property.
6. The test device leaves the session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* Getting initial values for all properties are succeeded.
* Setting initial values for all read-write properties are succeeded.
* Setting properties to valid value are succeeded.
  + The IsRecirculating property is set to *false*, and the PropertiesChanged signal is received.

## HAE-v1-03: AudioVideoInput Interface Test

Objective

Verify the properties, methods and signals of the AudioVideoInput interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is AudioVideoInput Interface on DUT.
3. Get initial values for all properties.
   * Retrieve the InputSourceId property.
   * Retrieve the SupportedInputSources property.
4. Initialize all read-write properties.
   * Initialize the InputSourceId property to the 1st item of the SupportedInputSources.
5. Set properties to invalid value.
   * Set the InputSourceId property to an invalid value.
     + Get the InputSourceId property.
6. Set properties to valid value.
   * If *size of SupportedInputSources!=1*, set the InputSourceId property to *2nd item of SupportedInputSources*.
     + Wait the PropertiesChanged signal for the InputSourceId property.
     + Get the InputSourceId property.
7. The test device leaves the session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* Getting initial values for all properties are succeeded.
* Setting initial values for all read-write properties are succeeded.
* The result of setting properties with invalid value is failed.
  + Setting the InputSourceId property to an invalid value is failed.
  + The current InputSourceId is 1st item of SupportedInputSources.
* Setting properties to valid value are succeeded.
  + If *size of SupportedInputSources != 1*, the InputSourceId property is set to *2nd item of SupportedInputSources*, and the PropertiesChanged signal is received.

## HAE-v1-04: AudioVolume Interface Test

Objective

Verify the properties, methods and signals of the AudioVolume interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is AudioVolume Interface on DUT.
3. Get initial values for all properties.
   * Retrieve the Volume property.
   * Retrieve the MaxVolume property.
   * Retrieve the Mute property.
4. Initialize all read-write properties.
   * Initialize the Volume property to *0*.
   * Initialize the Mute property to *false*.
5. Set properties to valid value.
   * Set the Volume property tovalue*1*.
     + Wait the PropertiesChanged signal for the Volume property.
     + Get the Volume property.
   * Set the Mute property to *true*.
     + Wait the PropertiesChanged signal for the Mute property.
     + Get the Mute property.
6. Set properties to invalid value.
   * If *MaxVolume != UINT8\_MAX*, Set the Volume property to *MaxVolume + 1.*
     + Wait the PropertiesChanged signal for the Volume property.
     + Get the Volume property.
7. The test device leaves the session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* Getting initial values for all properties are succeeded.
* Setting initial values for all read-write properties are succeeded.
* Setting properties to valid value are succeeded.
  + The Volume property is set to *1*, and the PropertiesChanged signal is received.
  + The Mute property is set to *true*, and the PropertiesChanged signal is received.
* The result of setting properties to invalid value is as follows.
  + If *MaxVolume != 1*, The Volume property is set to *MaxVolume* and the PropertiesChanged signal is received.

## HAE-v1-05: BatteryStatus Interface Test

Objective

Verify the properties, methods and signals of the BatteryStatus interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is BatteryStatus Interface on DUT.
3. Get initial values for all properties.
   * Retrieve the CurrentValue property.
   * Retrieve the IsCharging property.
4. The test device leaves session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* Getting initial value for all properties are succeeded.

## HAE-v1-06: Channel Interface Test

Objective

Verify the properties, methods and signals of the Channel interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is Channel Interface on DUT.
3. Get initial values for all properties.
   * Retrieve the ChannelId property.
   * Retrieve the TotalNumberOfChannels property.
   * Calls GetChannelList method with following.
     + startingRecord = *0*
     + numRecords =*min(100, TotalNumberOfChannels)*
4. Initialize all read-write properties.
   * Initialize the ChannelId property to the *1st item of the ChannelList*.
5. Set properties to invalid value.
   * Set the ChannelId property to *“Invalid\_ChannelId”.*
     + Get the ChannelId property.
6. Set properties to valid value.
   * If TotalNumberOfChannels>*1*, set the ChannelId property to the *2nd item of the ChannelList*.
     + Wait the PropertiesChanged signal for the Channel property.
     + Get the Channel property.
7. Call method with invalid params
   * Call the GetChannelList method with invalid startingRecord param as following.
     + startingRecord = *TotalNumberOfChannels + 1*
     + numRecords = *50*
   * Call the GetChannelList method with invalid numRecords param as following.
     + startingRecord = *max(0, TotalNumberOfChannels – 50)*
     + numRecords = *100*
8. Call method with valid params.
   * Call the GetChannelList method with valid startingRecord param as following.
     + startingRecord = *0*
     + numRecords = *min(100, TotalNumberOfChannels)*
9. The test device leaves the session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* Getting initial values for all properties are succeeded.
* Setting initial values for all read-write properties are succeeded.
* Setting properties to invalid value are failed.
  + Set the ChannelId property to *“Invalid\_ChannelId”* is failed, and the current ChannelId keeps the *1st item of ChannelList*.
* Setting properties to valid value are succeeded.
  + The ChannelId property is set to the *2nd item of ChannelList*, and the PropertiesChanged signal is received.
* The result of method call with invalid params is as following:
  + The GetChannelList method call with the invalid startingRecord param is failed, and returns “org.alljoyn.Error.InvalidValue” error.
  + The GetChannelList method call with invalid numRecords param is succeeded and returns a channel list that has TotalNumberOfChannels – *startingRecord* size.
* The result of method call with valid params is as following:
  + *Min(100, TotalNumberOfChannels)* channels are returned from *1st item of ChannelList.*

## HAE-v1-07: ClimateControlMode Interface Test

Objective

Verify the properties, methods and signals of the ClimateControlMode interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is ClimateControlMode Interface on DUT.
3. Get initial values for all properties.
   * Retrieve the Mode property.
   * Retrieve the SupportedModes property.
   * Retrieve the OperationalState property.
4. Initialize all read-write properties.
   * Initialize the Mode property to the *1st item of SupportedModes*.
5. Set properties to invalid value.
   * Set the Mode property to *0xFF*(not supported).
     + Get the Mode property.
6. Set properties to valid value.
   * If size of SupportedModes>*1*, set the Mode property to the *2nd item of the SupportedModes*.
     + Wait the PropertiesChanged signal for the Mode property.
     + Get the Mode property
7. The test device leaves the session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* Getting initial values for all properties are succeeded.
* Setting initial values for all read-write properties are succeeded
* Setting properties to invalid value are failed.
  + Setting the Mode property to *0xFF*(not supported) is failed, and the current Mode is *1st item of the SupportedModes*.
* Setting properties to valid value are succeeded.
  + If *size of SupportedModes> 1*, the Mode property is *2nd item of the SupportedModes* and the PropertiesChanged signal is received.

## HAE-v1-08: ClosedStatus Interface Test

Objective

Verify the properties, methods and signals of the ClosedStatus interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is ClosedStatus Interface on DUT.
3. Get initial values for all properties.
   * Retrieve the IsClosed property.
4. The test device leaves session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* Getting initial value for all properties is succeeded.

## HAE-v1-09: CurrentPower Interface Test

Objective

Verify the properties, methods and signals of the CurrentPower interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is CurrentPower Interface on DUT.
3. Get initial values for all properties.
   * Retrieve the CurrentPower property.
   * Retrieve the Precision property.
   * Retrieve the UpdateMinTime property.
4. The test device leaves the session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* Getting initial values for all properties are succeeded.

## HAE-v1-10: CurrentTemperature Interface Test

Objective

Verify the properties, methods and signals of the CurrentTemperature interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is CurrentTemperature Interface on DUT.
3. Get initial values for all properties.
   * Retrieve the CurrentValue property.
   * Retrieve the Precision property.
   * Retrieve the UpdateMinTime property.
4. The test device leaves the session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* Getting initial values for all properties are succeeded.

## HAE-v1-11: CycleControl Interface Test

Objective

Verify the properties, methods and signals of the CycleControl interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is CycleControl Interface on DUT.
3. Get initial values for all properties.
   * Retrieve the OperationalState property.
   * Retrieve the SupportedOperationalStates property.
   * Retrieve the SupportedOperationalCommands property.
4. Call method with invalid params
   * Call the ExecuteCommand method with invalid value “*UnsupportedCommand*”.
5. Call method with valid params.
   * Call the ExecuteCommand method with 1st item of the SupportedOperationalCommands.
6. The test device leaves the session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* Getting initial values for all properties are succeeded.
* The result of method call with invalid params is failed and returns “*org.alljoyn.Error.InvalidValue*” error.
* The result of method call with valid params is succeeded.

## HAE-v1-12: DishWashingCyclePhase Interface Test

Objective

Verify the properties, methods and signals of the DishWashingCyclePhase interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is DishWashingCyclePhase Interface on DUT.
3. Get initial values for all properties.
   * Retrieve the CyclePhase property.
   * Retrieve the SupportedCyclePhases property.
4. Call method with invalid param.
   * Call the GetVendorPhasesDescription method with *“Invalid\_lang”.*
5. Call method with valid param.
   * Call the GetVendorPhasesDescription method with *“en”*.
6. The test device leaves the session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* Getting initial values for all properties are succeeded.
* Method call with invalid param is failed and returns “*org.alljoyn.Error.LanguageNotSupported*” error.
* Method call with valid param is succeeded.

## HAE-v1-13: EnergyUsage Interface Test

Objective

Verify the properties, methods and signals of the EnergyUsage interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is EnergyUsage Interface on DUT.
3. Get initial values for all properties.
   * Retrieve the CumulativeEnergy property.
   * Retrieve the Precision property.
   * Retrieve the UpdateMinTime property.
4. Call method.
   * Call the ResetCumulativeEnergy method.
     + If *CumulativeEnergy != 0.0*, wait the PropertiesChanged signal for the ResetCumulativeEnergy property.
     + Get the CumulativeEnergy property.
5. The test device leaves the session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* Getting initial values for all properties are succeeded.
  + The ResetCumulativeEnergy method call is succeeded. The CumulativieEnergy property is set to *0*, and the PropertiesChanged signal is received.

## HAE-v1-14: FanSpeedLevel Interface Test

Objective

Verify the properties, methods and signals of the FanSpeedLevel interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is FanSpeedLevel Interface on DUT.
3. Get initial values for all properties.
   * Retrieve the FanSpeedLevel property.
   * Retrieve the MaxFanSpeedLevel property.
   * Retrieve the AutoMode property.
4. Initialize all read-write properties.
   * Initialize the FanSpeedLevel property to *0x01*.
   * If AutoMode != *0xFF*(not supported), initialize AutoMode property to *0x00*(Off).
5. Set properties to invalid value.
   * If *MaxFanSpeedLevel != UINT8\_MAX*, set the FanSpeedLevel property to *MaxFanSpeedLevel + 1*.
     + Get the FanSpeedLevel property.
   * Set the FanSpeedLevel property to *0x00* (Off).
     + Get the FanSpeedLevel property.
   * Set the AutoMode property to *0xFF*(not supported).
     + Get the AutoMode property.
   * Set the AutoMode property to *0x02*.
     + Get the AutoMode property.
6. Set properties to valid value
   * If *MaxFanSpeedLevel> 1*, set the FanSpeedLevel property to*2*.
     + Wait the PropertiesChanged signal for the FanSpeedLevel property.
     + Get the FanSpeedLevel property.
   * If AutoMode != *0xFF*, set the AutoMode property to *0x01.*
     + Wait the PropertiesChanged signal for the AutoMode property.
     + Get the AutoMode property.
7. The test device leaves the session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* Getting initial values for all properties are succeeded.
* Setting initial values for all read-write properties are succeeded.
* Setting properties to invalid value are failed.
  + If *MaxFanSpeedLevel != UINT8\_MAX*, Setting the FanSpeedLevel property to *MaxFanSpeedLevel + 1* is failed and the current FanSpeedLevel property is *0x01*.
  + Setting the FanSpeedLevel property to *0x00*(Off) is failed and the FanSpeedLevel property is *0x01*.
  + Setting the property to *0xFF*(not supported) is failed.
    - If *AutoMode != 0xFF* then the current AutoMode is *0x00*.
    - Else the current AutoMode is *0xFF*.
  + Setting the AutoMode property to *0x02* is failed.
    - If AutoMode != 0xFF then the current AutoMode is *0x00*.
    - Else the current AutoMode is *0xFF.*
* Setting properties to valid value are succeeded.
  + If *MaxFanSpeedLevel> 1*, the FanSpeedLevel property is set to 2, and the PropertiesChanged signal is received.
    - The current FanSpeedLevel property is 2.
  + If *AutoMode != 0xFF*, the AutoMode property is set to 0x01 and the PropertiesChanged signal is received.
    - The current AutoMode is 0x01.

## HAE-v1-15: HeatingZone Interface Test

Objective

Verify the properties, methods and signals of the HeatingZone interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is HeatingZone Interface on DUT.
3. Get initial values for all properties.
   * Retrieve the NumberOfHeatingZones property.
   * Retrieve the MaxHeatingLevels property.
   * Retrieve the HeatingLevels property.
4. The test device leaves session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* Getting initial value for all properties is succeeded.

## HAE-v1-16: Hid Interface Test

Objective

Verify the properties, methods and signals of the Hid interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is Hid Interface on DUT.
3. Get initial values for all properties.
   * Retrieve the SupportedEvents property.
4. Call method
   * If *size of SupportedEvents> 1*, call the InjectEvents method with 1st item of SupportedInputEvents.
5. The test device leaves the session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* Getting initial values for all properties are succeeded.
* The InjectEvents method call is succeeded.

## HAE-v1-17: LaundryCyclePhase Interface Test

Objective

Verify the properties, methods and signals of the LaundryCyclePhase interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is LaundryCyclePhase Interface on DUT.
3. Get initial values for all properties.
   * Retrieve the CyclePhase property.
   * Retrieve the SupportedCyclePhases property.
4. Call method with invalid param.
   * Call the GetVendorPhasesDescription method with *“Invalid\_lang”.*
5. Call method with valid param.
   * Call the GetVendorPhasesDescription method with *“en”*.
6. The test device leaves the session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* Getting initial values for all properties are succeeded.
* Method call with invalid param is failed and returns “*org.alljoyn.Error.LanguageNotSupported*” error.
* Method call with valid param is succeeded.

## HAE-v1-18: OffControl Interface Test

Objective

Verify the properties, methods and signals of the OffControl interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is OffControl Interface on DUT.
3. Call method.
   * Call the SwitchOff method.
4. The test device leaves the session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* The SwitchOff method call is succeeded.

## HAE-v1-19: OnControl Interface Test

Objective

Verify the properties, methods and signals of the OnControl interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is OnControl Interface on DUT.
3. Call method.
   * Call the SwitchOn method.
4. The test device leaves the session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* The SwitchOn method call is succeeded.

## HAE-v1-20: OnOffStatus Interface Test

Objective

Verify the properties, methods and signals of the OnOffStatus interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is OnOffStatus Interface on DUT.
3. Get initial values for all properties.
   * Retrieve the OnOff property.
4. The test device leaves the session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* Getting initial values for all properties is succeeded.

## HAE-v1-21: OvenCyclePhase Interface Test

Objective

Verify the properties, methods and signals of the OvenCyclePhase interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is OvenCyclePhase Interface on DUT.
3. Get initial values for all properties.
   * Retrieve the CyclePhase property.
   * Retrieve the SupportedCyclePhases property.
4. Call method with invalid param.
   * Call the GetVendorPhasesDescription method with *“Invalid\_lang”.*
5. Call method with valid param.
   * Call the GetVendorPhasesDescription method with *“en”*.
6. The test device leaves the session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* Getting initial values for all properties are succeeded.
* Method call with invalid param is failed and returns “*org.alljoyn.Error.LanguageNotSupported*” error.
* Method call with valid param is succeeded.

## HAE-v1-22: RapidMode Interface Test

Objective

Verify the properties, methods and signals of the RapidMode interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is RapidMode Interface on DUT.
3. Get initial values for all properties.
   * Retrieve the RapidMode property.
4. Initialize all read-write properties.
   * Initialize the RapidMode to false.
5. Set properties to valid value.
   * Set the RapidMode to true.
     + Wait the PropertiesChanged signal for RapidMode property.
     + Get the RapidMode property.
6. The test device leaves session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* Getting initial value for all properties are succeeded.
* Setting initial values for all read-write properties are succeeded
* Setting properties to valid value are succeeded.
  + The RapidMode is set to true, and the PropertiesChanged signal is received.

## HAE-v1-23: RemoteControllability Interface Test

Objective

Verify the properties, methods and signals of the RemoteControllaibily interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is RemoteControllaibily Interface on DUT.
3. Get initial values for all properties.
   * Retrieve the IsControllable property.
4. The test device leaves session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* Getting initial value for all properties is succeeded.

## HAE-v1-24: RepeatMode Interface Test

Objective

Verify the properties, methods and signals of the RepeatMode interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is RepeatMode Interface on DUT.
3. Get initial values for all properties.
   * Retrieve the RepeatMode property.
4. Initialize all read-write properties.
   * Initialize the RepeatMode property to false.
5. Set properties to valid value.
   * Set the RepeatMode property to true.
     + Wait the PropertiesChanged signal for RepeatMode property
     + Get the RepeatMode property
6. The test device leaves the session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* Getting initial values for all properties are succeeded.
* Setting initial values for all read-write properties are succeeded.
* Setting properties to valid value are succeeded.
  + The RepeatMode property is set to true, and the PropertiesChanged signal is received.

## HAE-v1-25: ResouceSaving Interface Test

Objective

Verify the properties, methods and signals of the ResouceSaving interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is ResouceSaving Interface on DUT.
3. Get initial values for all properties.
   * Retrieve the ResourceSavingMode property.
4. Initialize all read-write properties.
   * Initialize the ResourceSavingMode property to false.
5. Set properties to valid value.
   * Set the ResourceSavingMode property to true.
     + Wait the PropertiesChanged signal for ResourceSavingMode property.
     + Get the ResourceSavingMode property.
6. The test device leaves the session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* Getting initial values for all properties are succeeded.
* Setting initial values for all read-write properties are succeeded
* Setting properties to valid value are succeeded.
  + The ResourceSavingMode property is set to true, and the PropertiesChanged signal is received.

## HAE-v1-26: RobotCleaningCyclePhase Interface Test

Objective

Verify the properties, methods and signals of the RobotCleaningCyclePhase interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is RobotCleaningCyclePhase Interface on DUT.
3. Get initial values for all properties.
   * Retrieve the CyclePhase property.
   * Retrieve the SupportedCyclePhases property.
4. Call method with invalid param.
   * Call the GetVendorPhasesDescription method with *“Invalid\_lang”.*
5. Call method with valid param.
   * Call the GetVendorPhasesDescription method with *“en”*.
6. The test device leaves the session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* Getting initial values for all properties are succeeded.
* Method call with invalid param is failed and returns “*org.alljoyn.Error.LanguageNotSupported*” error.
* Method call with valid param is succeeded.

## HAE-v1-27: SoilLevel Interface Test

Objective

Verify the properties and signals of the SoilLevel interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is SoilLevel Interface on DUT.
3. Get initial values for all properties.
   * Retrieve the MaxLevel property.
   * Retrieve the TargetLevel property.
   * Retrieve the SelectableLevels property.
4. Initialize all read-write properties.
   * Initialize the TargetLevel property to the 0.
5. Set properties to valid value.
   * Set the TargetLevel property to “ValueFromSelectableLevels”.
     + Wait the PropertiesChanged signal for the TargetLevel property.
     + Get the TargetLevel property.
6. Set properties to invalid value
   * Set the TargetLevel property to MaxLevel + 1.
     + Get the TargetLevel property.
   * Set the TargetLevel property to “ValueOutsideSelectableLevels”.
     + Get the TargetLevel property.
7. The test device leaves the session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* Getting initial values for all properties is succeeded.
* Setting initial values for all read-write properties is succeeded.
* Setting properties to valid value is succeeded:
  + The TargetLevel is set to “ValueFromSelectableLevels”, and the PropertiesChanged signal is received.
* The result of setting properties to invalid value is as follows.
  + Set the TargetLevel property to *MaxLevel+1* is failed, and the current TargetLevel keeps the *“*ValueFromSelectableLevels*”*
  + Set the TargetLevel property to “ValueOutsideSelectableLevels” is failed, and the current TargetLevel keeps the *“*ValueFromSelectableLevels”

## HAE-v1-28: SpinSpeedLevel Interface Test

Objective

Verify the properties and signals of the SpinSpeedLevel interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is SpinSpeedLevel Interface on DUT.
3. Get initial values for all properties.
   * Retrieve the MaxLevel property.
   * Retrieve the TargetLevel property.
   * Retrieve the SelectableLevels property.
4. Initialize all read-write properties.
   * Initialize the TargetLevel property to the 0.
5. Set properties to valid value.
   * Set the TargetLevel property to “ValueFromSelectableLevels”.
     + Wait the PropertiesChanged signal for the TargetLevel property.
     + Get the TargetLevel property.
6. Set properties to invalid value
   * Set the TargetLevel property to MaxLevel + 1.
     + Get the TargetLevel property.
   * Set the TargetLevel property to “ValueOutsideSelectableLevels”.
     + Get the TargetLevel property.
7. The test device leaves the session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* Getting initial values for all properties is succeeded.
* Setting initial values for all read-write properties is succeeded.
* Setting properties to valid value is succeeded:
  + The TargetLevel is set to “ValueFromSelectableLevels”, and the PropertiesChanged signal is received.
* The result of setting properties to invalid value is as follows.
  + Set the TargetLevel property to *MaxLevel+1* is failed, and the current TargetLevel keeps the *“*ValueFromSelectableLevels*”*
  + Set the TargetLevel property to “ValueOutsideSelectableLevels” is failed, and the current TargetLevel keeps the *“*ValueFromSelectableLevels”

## HAE-v1-29: TargetTemperature Interface Test

Objective

Verify the properties and signals of the TargetTemperature interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is TargetTemperature Interface on DUT.
3. Get initial values for all properties.
   * Retrieve the TargetValue property.
   * Retrieve the MinValue property.
   * Retrieve the MaxValue property.
   * Retrieve the StepValue property.
4. Initialize all read-write properties.
   * Initialize the TargetValue property to the MinValue.
5. Set properties to valid value.
   * Set the TargetValue property to MaxValue.
     + Wait the PropertiesChanged signal for the TargetValue property.
     + Get the TargetValue property.
   * If StepValue != 0, Set the TargetValue property to MinValue + StepValue.
     + Wait the PropertiesChanged signal for the TargetValue property.
     + Get the TargetValue property.
   * Else, Set the TargetValue property to MinValue + (MaxValue - MinVaule)/2.
     + Wait the PropertiesChanged signal for the TargetValue property.
     + Get the TargetValue property.
6. Set properties to invalid value
   * If MinValue> DBL\_MIN, Set the TargetValue property to the DBL\_MIN.
     + Wait the PropertiesChanged signal for the TargetValue property.
     + Get the TargetValue property.
   * If MaxValue< DBL\_MAX, Set the TargetValue property to the DBL\_MAX.
     + Wait the PropertiesChanged signal for the TargetValue property.
     + Get the TargetValue property.
   * If StepValue != 0, Set the TargetValue property to MinValue + (StepValue/3)
     + Wait the PropertiesChanged signal for the TargetValue property.
     + Get the TargetValue property.
7. The test device leaves the session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* Getting initial values for all properties are succeeded.
* Setting initial values for all read-write properties are succeeded.
* Setting properties to valid value are succeeded.
  + The TargetValue property is set to MaxValue, and the PropertiesChanged signal is received.
  + If StepValue!= 0, the TargetValue property is set to MinValue + StepValue and the PropertiesChanged signal is received.
  + Else, the TargetValue property is set to MinValue + (MaxValue - MinVaule)/2and the PropertiesChanged signal is received.
* The result of setting properties to invalid value is as follows.
  + If MinValue> DBL\_MIN, the TargetValue property is set to MinValue and the PropertiesChanged signal is received.
  + If MaxValue< DBL\_MAX, the TargetValue property is set to MaxValue and the PropertiesChanged signal is received.
  + If StepValue != 0, the TargetValue property is set to MinValue or ‘MinValue + StepValue, and the PropertiesChanged signal is received.

## HAE-v1-30: Timer Interface Test

Objective

Verify the properties, methods and signals of the Timer interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an about announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is Timer Interface on DUT.
3. Get initial values for all properties.
   * Retrieve the ReferenceTimer property.
   * Retrieve the TargetTimeToStart property.
   * Retrieve the TargetTimeToStop property.
   * Retrieve the EstimatedTimeToEnd property.
   * Retrieve the RunningTime property.
   * Retrieve the TargetDuration property.
4. Call method with params
   * Call the SetTargetTimeToStart method with value *targetTimeToStart*.
     + Wait the PropertiesChanged signal for the TargetTimeToStart property.
     + Get the TargetTimeToStart property.
   * Call the SetTargetTimeToStop method with value *targetTimeToStop*.
     + Wait the PropertiesChanged signal for the TargetTimeToStop property.
     + Get the TargetTimeToStop property.
5. The test device leaves the session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* Getting initial values for all properties are succeeded.
* The result of method call with params
  + Method call of SetTargetTimeToStart method is succeeded.
  + The TargetTimeToStart is set to *targetTimeToStart* value
  + Method call of SetTargetTimeToStop method is succeeded.
  + The *TargetTimeToStop* is set to *targetTimeToStop* value

## HAE-v1-31: WaterLevel Interface Test

Objective

Verify the properties and signals of the WaterLevel interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is WaterLevel Interface on DUT.
3. Get initial values for all properties.
   * Retrieve the SupplySource property.
   * Retrieve the MaxLevel property.
   * Retrieve the CurrentLevel property.
4. The test device leaves the session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* Getting initial values for all properties is succeeded.

## HAE-v1-32: WindDirection Interface Test

Objective

Verify the properties, methods and signals of the WindDirection interface of DUT.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application at the port specified in the received About announcement if there is WindDirection Interface on DUT.
3. Get initial values for all properties.
   * Retrieve the HorizontalDirection property.
   * Retrieve the HorizontalMax property.
   * Retrieve the HorizontalAutoMode property.
   * Retrieve the VerticalDirection property.
   * Retrieve the VerticalMax property.
   * Retrieve the VerticalAutoMode property.
4. Initialize all read-write properties.
   * Initialize the HorizontalDirection property to 0.
   * Initialize the HorizontalAutoMode property to 0(Off).
   * Initialize the VerticalDirection property to 0.
   * Initialize the VerticalAutoMode property to 0(Off).
5. Set properties to valid value
   * Set the HorizontalDirection property to 1.
     + Wait the PropertiesChanged signal for the HorizontalDirection property.
     + Get the HorizontalDirection property.
   * If HorizontalMax != 1, Set the HorizontalDirection property to HorizontalMax
     + Wait the PropertiesChanged signal for the HorizontalDirection property.
     + Get the HorizontalDirection property.
   * If HorizontalAutoMode != 0xFF, Set the HorizontalAutoMode property to 1(On).
     + Wait the PropertiesChanged signal for the HorizontalAutoMode property.
     + Get the HorizontalAutoMode property.
   * Set the VerticalDirection property to 1.
     + Wait the PropertiesChanged signal for the VerticalDirection property.
     + Get the VerticalDirection property.
   * If VerticalMax != 1, Set the VerticalDirection property to VerticalMax.
     + Wait the PropertiesChanged signal for the VerticalDirection property.
     + Get the VerticalDirection property.
   * If VerticalAutoMode != 0xFF, Set the VerticalAutoMode property to 1(On).
     + Wait the PropertiesChanged signal for the VerticalAutoMode property.
     + Get the VerticalAutoMode property.
6. Set properties to invalid value.
   * If HorizontalMax != UINT16\_MAX, Set the HorizontalDirection property to HorizontalMax + 1.
     + Get the HorizontalDirection property.
   * Set the HorizontalAutoMode property to 2.
     + Get theHorizontalAutoMode property.
   * If VerticalMax != UINT16\_MAX, Set the VerticalDirection property to VerticalMax + 1.
     + Get the VerticalDirection property.
   * Set the VerticalAutoMode property to 2.
     + Get the VerticalAutoMode property.
7. The test device leaves the session.

Expected results

* The test device receives an About announcement from the application on the DUT.
* The test device joins a session with the application at the port specified in the received About announcement.
* Getting initial values for all properties are succeeded.
* Setting initial values for all read-write properties are succeeded
* Setting properties to valid value are succeeded.
  + The HorizontalDirection property is set to 1, and the PropertiesChanged signal is received.
  + If HorizontalMax != 1, theHorizontalDirection property is set to HorizontalMax and the PropertiesChanged signal is received.
  + If HorizontalAutoMode != 0xFF, the HorizontalAutoMode property is set to 1(On) and the PropertiesChanged signal is received.
  + The VerticalDirection property is set to 1, and the PropertiesChanged signal is received.
  + If VerticalMax != 1, the VerticalDirection property is set to VerticalMax and the PropertiesChanged signal is received.
  + If VerticalAutoMode != 0xFF, the VerticalAutoMode property is set to 1(On) and the PropertiesChanged signal is received.
* The result of setting properties to invalid value is as follows.
  + If HorizontalMax != UINT16\_MAX, the HorizontalDirection property is set to HorizontalMax and the PropertiesChanged signal is received.
  + Setting the HorizontalAutoMode property to 2 is failed and current HorizontalAutoMode keeps 1.
  + If VerticalMax != UINT16\_MAX, VerticalDirection property is set to VerticalMax and the PropertiesChanged signal is received.
  + Setting theVerticalAutoMode property to 2 is failed and current VerticalAutoMode keeps 1.

1. Implementation Conformance Statement, ICS

The purpose of this ICS proforma is to provide a mechanism whereby a supplier of an implementation of the requirements defined in relevant specifications may provide information about the implementation in a standardized manner. The supplier of the implementation shall complete the ICS proforma in each of the spaces provided. The ICS proforma contained in this appendix (tables below) is comprised of information in tabular form and includes following columns:

Item column

The item column contains a number which identifies the item in the table.

Description column

The item description column describes in free text each respective item (e.g. parameters, timers, etc.). It implicitly means "is supported by the implementation?".

Status column

The following notations are used for the status column:

|  |  |
| --- | --- |
| M | mandatory – the capability is required to be supported |
| O | optional – the capability may be supported or not. |
| N/A | not applicable – in the given context, it is impossible to use the capability. |
| C.i | conditional – the requirement on the capability ("M", "O" or "N/A") depends on the support of other optional or conditional items. "i" is an integer identifying an unique conditional status expression which is defined immediately following the table. For nested conditional expressions, the syntax "IF ... THEN (IF ... THEN ... ELSE...) ELSE ..." shall be used to avoid ambiguities. |

Support column

The support column shall be filled in by the supplier of the implementation. The following common notations are used for the support column:

|  |  |
| --- | --- |
| Y or y | supported by the implementation |
| N or n | not supported by the implementation |
| N/A, n/a or - | N/A, n/a or - no answer required (allowed only if the status is N/A, directly or after evaluation of a conditional status) |

Mnemonic column

The Mnemonic column contains mnemonic identifiers for each item.

**Table A.1: HAE ICS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Description** | **Status** | **Support** | **Mnemonic** |
| 1 | Support of Hae Service Framework | O |  | ICSHAE\_HaeServiceFramework |
| 2 | Support of AirRecirculationMode Interface | C.01 |  | ICSHAE\_AirRecirculationModeInterface |
| 3 | Support of AudioVideoInput Interface | C.01 |  | ICSHAE\_AudioVideoInputInterface |
| 4 | Support of AudioVolume Interface | C.02 |  | ICSHAE\_AudioVolumeInterface |
| 5 | Support of BatteryStatus Interface | C.01 |  | ICSHAE\_BatteryStatusInterface |
| 6 | Support of Channel Interface | C.02 |  | ICSHAE\_ChannelInterface |
| 7 | Support of ClimateControlMode Interface | C.03 |  | ICSHAE\_ClimateControlModeInterface |
| 8 | Support of ClosedStatus Interface | C.04 |  | ICSHAE\_ClosedStatusInterface |
| 9 | Support of CurrentPower Interface | C.01 |  | ICSHAE\_CurrentPowerInterface |
| 10 | Support of CurrentTemperature Interface | C.03 |  | ICSHAE\_CurrentTemperatureInterface |
| 11 | Support of CycleControl Interface | C.05 |  | ICSHAE\_CycleControlInterface |
| 12 | Support of DishWashingCyclePhase Interface | C.01 |  | ICSHAE\_DishWashingCyclePhaseInterface |
| 13 | Support of EnergyUsage Interface | C.01 |  | ICSHAE\_EnergyUsageInterface |
| 14 | Support of FanSpeedLevel Interface | C.06 |  | ICSHAE\_FanSpeedLevelInterface |
| 15 | Support of HeatingZone Interface | C.07 |  | ICSHAE\_HeatingZoneInterface |
| 16 | Support of Hid Interface | C.01 |  | ICSHAE\_HidInterface |
| 17 | Support of LaundryCyclePhase Interface | C.01 |  | ICSHAE\_LaundryCyclePhaseInterface |
| 18 | Support of OffControl Interface | C.01 |  | ICSHAE\_OffControlInterface |
| 19 | Support of OnControl Interface | C.01 |  | ICSHAE\_OnControlInterface |
| 20 | Support of OnOffStatus Interface | C.08 |  | ICSHAE\_OnOffStatusInterface |
| 21 | Support of OvenCyclePhase Interface | C.01 |  | ICSHAE\_OvenCyclePhaseInterface |
| 22 | Support of RapidMode Interface | C.01 |  | ICSHAE\_RapidModeInterface |
| 23 | Support of RemoteControllability Interface | C.01 |  | ICSHAE\_RemoteControllabilityInterface |
| 24 | Support of RepeatMode Interface | C.01 |  | ICSHAE\_RepeatModeInterface |
| 25 | Support of ResouceSaving Interface | C.01 |  | ICSHAE\_ResouceSavingInterface |
| 26 | Support of RobotCleaningCyclePhase Interface | C.01 |  | ICSHAE\_RobotCleaningCyclePhaseInterface |
| 27 | Support of SoilLevel Interface | C.01 |  | ICSHAE\_SoilLevelInterface |
| 28 | Support of SpinSpeedLevel Interface | C.01 |  | ICSHAE\_SpinSpeedLevelInterface |
| 29 | Support of TargetTemperature Interface | C.09 |  | ICSHAE\_TargetTemperatureInterface |
| 30 | Support of Timer Interface | C.01 |  | ICSHAE\_TimerInterface |
| 31 | Support of WaterLevel Interface | C.01 |  | ICSHAE\_WaterLevelInterface |
| 32 | Support of WindDirection Interface | C.01 |  | ICSHAE\_WindDirectionInterface |
| (DT : DeviceType in “DeviceTypeDescription” (mandatory) field of Hae AboutData)  C.01 IF (A.1/1) THEN O ELSE N/A  C.02 IF (A.1/1) THEN (IF (DT=21) THEN M ELSE O) ELSE N/A  C.03 IF (A.1/1) THEN (IF (DT=5 OR DT=6) THEN M ELSE O) ELSE N/A  C.04 IF (A.1/1) THEN (IF (DT=2 OR DT= 3) THEN M ELSE O) ELSE N/A  C.05 IF (A.1/1) THEN (IF (DT=12 OR DT=13 OR DT=14 OR DT=15 OR DT=16 OR DT=17) THEN M ELSE O) ELSE N/A  C.06 IF (A.1/1) THEN (IF (DT=18) THEN M ELSE O) ELSE N/A  C.07 IF (A.1/1) THEN (IF (DT=19) THEN M ELSE O) ELSE N/A  C.08 IF (A.1/1) THEN (IF (DT=5 OR DT=6 OR DT=7 OR DT=8 OR DT=9 OR DT=10 OR DT=12 OR DT=13 OR DT=14 OR DT=15 OR DT=16 OR DT=17 OR DT=18 OR DT=21 OR DT=22) THEN M ELSE O) ELSE N/A  C.09 IF (A.1/1) THEN (IF (DT=5 OR DT=6 OR DT=20) THEN M ELSE O) ELSE N/A | | | | |

1. Implementation eXtra Information for Testing, IXIT

**Table B.1: Hae features**

| **Item** | **Name** | **Description** | **Default Value** | **Mnemonic** | **Value** |
| --- | --- | --- | --- | --- | --- |
| 1 | AirRecirculationMode Version | AirRecirculationMode Interface version number | 1 | IXITL\_AirRecirculationModeVersion |  |
| 2 | AudioVideoInput Version | AudioVideoInput Interface version number | 1 | IXITL\_AudioVideoInputVersion |  |
| 3 | AudioVolume Version | AudioVolume Interface version number | 1 | IXITL\_AudioVolumeVersion |  |
| 4 | BatteryStatus Version | BatteryStatus Interface version number | 1 | IXITL\_BatteryStatusVersion |  |
| 5 | Channel Version | Channel Interface version number | 1 | IXITL\_ChannelVersion |  |
| 6 | ClimateControlMode Version | ClimateControlMode Interface version number | 1 | IXITL\_ClimateControlModeVersion |  |
| 7 | ClosedStatus Version | ClosedStatus Interface version number | 1 | IXITL\_ClosedStatusVersion |  |
| 8 | CurrentPower Version | CurrentPower Interface version number | 1 | IXITL\_CurrentPowerVersion |  |
| 9 | CurrentTemperature Version | CurrentTemperature Interface version number | 1 | IXITL\_CurrentTemperatureVersion |  |
| 10 | CycleControl Version | CycleControl Interface version number | 1 | IXITL\_CycleControlVersion |  |
| 11 | DishWashingCyclePhase Version | DishWashingCyclePhase Interface version number | 1 | IXITL\_DishWashingCyclePhaseVersion |  |
| 12 | EnergyUsage Version | EnergyUsage Interface version number | 1 | IXITL\_EnergyUsageVersion |  |
| 13 | FanSpeedLevel Version | FanSpeedLevel Interface version number | 1 | IXITL\_FanSpeedLevelVersion |  |
| 14 | HeatingZone Version | HeatingZone Interface version number | 1 | IXITL\_HeatingZoneVersion |  |
| 15 | Hid Version | Hid Interface version number | 1 | IXITL\_HidVersion |  |
| 16 | LaundryCyclePhase Version | LaundryCyclePhase Interface version number | 1 | IXITL\_LaundryCyclePhaseVersion |  |
| 17 | OffControl Version | OffControl Interface version number | 1 | IXITL\_OffControlVersion |  |
| 18 | OnControl Version | OnControl Interface version number | 1 | IXITL\_OnControlVersion |  |
| 19 | OnOffStatus Version | OnOffStatus Interface version number | 1 | IXITL\_OnOffStatusVersion |  |
| 20 | OvenCyclePhase Version | OvenCyclePhase Interface version number | 1 | IXITL\_OvenCyclePhaseVersion |  |
| 21 | RapidMode Version | RapidMode Interface version number | 1 | IXITL\_RapidModeVersion |  |
| 22 | RemoteControllability Version | RemoteControllability Interface version number | 1 | IXITL\_RemoteControllabilityVersion |  |
| 23 | RepeatMode Version | RepeatMode Interface version number | 1 | IXITL\_RepeatModeVersion |  |
| 24 | ResouceSaving Version | ResouceSaving Interface version number | 1 | IXITL\_ResouceSavingVersion |  |
| 25 | RobotCleaningCyclePhase Version | RobotCleaningCyclePhase Interface version number | 1 | IXITL\_RobotCleaningCyclePhaseVersion |  |
| 26 | SoilLevel Version | SoilLevel Interface version number | 1 | IXITL\_SoilLevelVersion |  |
| 27 | SpinSpeedLevel Version | SpinSpeedLevel Interface version number | 1 | IXITL\_SpinSpeedLevelVersion |  |
| 28 | TargetTemperature Version | TargetTemperature Interface version number | 1 | IXITL\_TargetTemperatureVersion |  |
| 29 | Timer Version | Timer Interface version number | 1 | IXITL\_TimerVersion |  |
| 30 | WaterLevel Version | WaterLevel Interface version number | 1 | IXITL\_WaterLevelVersion |  |
| 31 | WindDirection Version | WindDirection Interface version number | 1 | IXITL\_WindDirectionVersion |  |

1. Test Case Mapping Table, TCMT

The applicability of each individual test is identified in the table B.1.

The applicability of every test is formally expressed by the use of Boolean expression that are based on parameters (ICS) included in annex A of the present document.

The columns in Table B.1 have the following meaning:

Test case column

The Test case column indicates the test case number for each test case as described in the Control Panel Service Framework test case specification for which the applicability is identified.

Description column

The Title column indicates the title of each test case as described in the Control Panel Service Framework test case specification for which the applicability is identified.

Release column

The Release column indicates the earliest release from which each test case is applicable, except if otherwise stated of an individual test case.

Status column

The following notations are used for the Status column:

|  |  |
| --- | --- |
| A | applicable - the test is applicable. |
| O | optional – the capability may be supported or not. |
| N/A | not applicable – in the given context, the test case is not applicable. |
| Ci | conditional – the test is applicable ("A") or not ("N/A") depending on the support of other optional or conditional items. "i" is an integer identifying an unique conditional status expression which is defined immediately following the table. For nested conditional expressions, the syntax "IF ... THEN (IF ... THEN ... ELSE...) ELSE ..." is used to avoid ambiguities. |

Table C.1: Test Case Mapping Table

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | Description | Release | Status |
| HAE-v1-01 | HAEAboutData Test | 16.04 | C01 |
| HAE-v1-02 | AirRecirculationMode Interface Test | 16.04 | C02 |
| HAE-v1-03 | AudioVideoInput Interface Test | 16.04 | C03 |
| HAE-v1-04 | AudioVolume Interface Test | 16.04 | C04 |
| HAE-v1-05 | BatteryStatus Interface Test | 16.04 | C05 |
| HAE-v1-06 | Channel Interface Test | 16.04 | C06 |
| HAE-v1-07 | ClimateControlMode Interface Test | 16.04 | C07 |
| HAE-v1-08 | ClosedStatus Interface Test | 16.04 | C08 |
| HAE-v1-09 | CurrentPower Interface Test | 16.04 | C09 |
| HAE-v1-10 | CurrentTemperature Interface Test | 16.04 | C10 |
| HAE-v1-11 | CycleControl Interface Test | 16.04 | C11 |
| HAE-v1-12 | DishWashingCyclePhase Interface Test | 16.04 | C12 |
| HAE-v1-13 | EnergyUsage Interface Test | 16.04 | C13 |
| HAE-v1-14 | FanSpeedLevel Interface Test | 16.04 | C14 |
| HAE-v1-15 | HeatingZone Interface Test | 16.04 | C15 |
| HAE-v1-16 | Hid Interface Test | 16.04 | C16 |
| HAE-v1-17 | LaundryCyclePhase Interface Test | 16.04 | C17 |
| HAE-v1-18 | OffControl Interface Test | 16.04 | C18 |
| HAE-v1-19 | OnControl Interface Test | 16.04 | C19 |
| HAE-v1-20 | OnOffStatus Interface Test | 16.04 | C20 |
| HAE-v1-21 | OvenCyclePhase Interface Test | 16.04 | C21 |
| HAE-v1-22 | RapidMode Interface Test | 16.04 | C22 |
| HAE-v1-23 | RemoteControllability Interface Test | 16.04 | C23 |
| HAE-v1-24 | RepeatMode Interface Test | 16.04 | C24 |
| HAE-v1-25 | ResouceSaving Interface Test | 16.04 | C25 |
| HAE-v1-26 | RobotCleaningCyclePhase Interface Test | 16.04 | C26 |
| HAE-v1-27 | SoilLevel Interface Test | 16.04 | C27 |
| HAE-v1-28 | SpinSpeedLevel Interface Test | 16.04 | C28 |
| HAE-v1-29 | TargetTemperature Interface Test | 16.04 | C29 |
| HAE-v1-30 | Timer Interface Test | 16.04 | C30 |
| HAE-v1-31 | WaterLevel Interface Test | 16.04 | C31 |
| HAE-v1-32 | WindDirection Interface Test | 16.04 | C32 |
| C01 IF (A.1/1) THEN A ELSE N/A C02 IF (A.1/2) THEN A ELSE N/A C03 IF (A.1/3) THEN A ELSE N/A C04 IF (A.1/4) THEN A ELSE N/A C05 IF (A.1/5) THEN A ELSE N/A C06 IF (A.1/6) THEN A ELSE N/A C07 IF (A.1/7) THEN A ELSE N/A C08 IF (A.1/8) THEN A ELSE N/A C09 IF (A.1/9) THEN A ELSE N/A C10 IF (A.1/10) THEN A ELSE N/A C11 IF (A.1/11) THEN A ELSE N/A C12 IF (A.1/12) THEN A ELSE N/A C13 IF (A.1/13) THEN A ELSE N/A C14 IF (A.1/14) THEN A ELSE N/A C15 IF (A.1/15) THEN A ELSE N/A C16 IF (A.1/16) THEN A ELSE N/A C17 IF (A.1/17) THEN A ELSE N/A C18 IF (A.1/18) THEN A ELSE N/A C19 IF (A.1/19) THEN A ELSE N/A C20 IF (A.1/20) THEN A ELSE N/A C21 IF (A.1/21) THEN A ELSE N/A C22 IF (A.1/22) THEN A ELSE N/A C23 IF (A.1/23) THEN A ELSE N/A C24 IF (A.1/24) THEN A ELSE N/A C25 IF (A.1/25) THEN A ELSE N/A C26 IF (A.1/26) THEN A ELSE N/A C27 IF (A.1/27) THEN A ELSE N/A C28 IF (A.1/28) THEN A ELSE N/A C29 IF (A.1/29) THEN A ELSE N/A C30 IF (A.1/30) THEN A ELSE N/A C31 IF (A.1/31) THEN A ELSE N/A C32 IF (A.1/32) THEN A ELSE N/A | | | |