AllJoyn™ Smart Home Service Framework 1.0 Test Case Specification

December 5, 2014

Contents

[1 Introduction 3](#_Toc404178426)

[1.1 Purpose 3](#_Toc404178427)

[1.2 Scope 3](#_Toc404178428)

[1.3 References 3](#_Toc404178429)

[2 Environment setup 4](#_Toc404178430)

[2.1 Requirements 4](#_Toc404178431)

[2.2 Preconditions 4](#_Toc404178432)

[2.3 Parameters 4](#_Toc404178433)

[3 Smart Home service framework test cases 5](#_Toc404178434)

[3.1 Smart home-v1-01: Service Interface Version equals 1 5](#_Toc404178435)

[3.2 Smart home-v1-02: ApplianceRegistration() 5](#_Toc404178436)

[3.3 Smart home-v1-03: Execute() 6](#_Toc404178437)

[3.4 Smart home-v1-04: ApplianceUnRegistration() 7](#_Toc404178438)

[3.5 Smart home-v1-05: DeviceHeartBeat () 8](#_Toc404178439)

# Introduction

## Purpose

These test cases evaluate and verify the functionality related to the AllJoyn™ Smart Home service framework exposed by a device through the Smart Home 1.0 interface.

The interface provides the centralized management capability to allow for a smart home client application to manage home appliances through the smart home server.

## Scope

These test cases are designed to determine if a device conforms to the Smart Home 1.0 interface specification. 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™Smart Home Service Framework 1.0 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 Smart Home interface according to the Smart Home service framework 1.0 interface specification.
* 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 process on the DUT is announcing its capabilities through its About announcement, including its support for the Smart Home interface
* Note that the appliance to be controlled is simulated by the test device in this test case specification.

## Parameters

Table 1. Parameters for the Smart Home service framework

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

# Smart Home service framework test cases

## Smart home-v1-01: Service Interface Version equals 1

Objective

Verify that the Service Interface Version of the DUT’s System App (the application supporting the Smart Home Service interface) is equal to 1.

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.
3. The test device calls the getVersion() property on the CentralizedManagement object.
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 interface version returned from the getVersion property equals 1.
* If not equal to 1, the test fails
* If a bus exception occurs, the test fails.
* Otherwise, the test case passes.

## Smart home-v1-02: ApplianceRegistration() method call

Objective

Verify that the test device can call the ApplianceRegistration() method to join the centralized management system.

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 with the port specified in the received About announcement.
3. The test device calls the ApplianceRegistration() method on the DUT’s centralized management object with the registration parameters (i.e., well known name , unique name and device Id).
4. The test device waits to receive the ReturnValue signal sent from the DUT.
5. The test device calls the Verification() method on the DUT’s centralized management object with device Id to verify whether the test device registers on the DUT successfully.
6. The test device waits to receive the returned parameters (i.e., well known name, unique name and device Id) according to the device Id from the DUT.
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 with the port specified in the received About announcement.
* The test device calls the ApplianceRegistration() method on the DUT’s centralized management object to provide the registration parameters (i.e., well known name, unique name and device Id) and the method returns a OK response.
* The test device receives the ValidateCode included in the ReturnValue signal sent from the DUT.
* The test device calls the Verification() method on the DUT’s centralized management object with device Id.
* The received parameters returned from the Verification() method on the DUT’s centralized management object is the same as the test device’s parameters (i.e., well known name , unique name and device Id) according to the device Id provided by the test device.

## Smart home-v1-03: Execute() method call

**Objective**

Verify that the test device can call the Execute() method to call the method of the appliance through the DUT. Note that the appliance to be controlled is simulated by the test device in this test case.

**Procedure**

1. The test device listens for an About announcement t from the application on the DUT.
2. After receiving an About announcement from the application, the test device joins a session with the application with the port specified in the received an About announcement.
3. The test device calls the Execute() method on the DUT’s centralized management object with isReturn, deviceId, objectPath, interfaceName, methodName (i.e., ApplianceControl() method) and method arguments maintained with the validation test.
4. The test device waits for the DUT to call the applianceControl() method of the test device with interfaceName, methodName , arguments that is provided in the Execute() method call.
5. The test device sends the replied value of the applianceControl() method call to the DUT.
6. The test device waits to receive the ReturnValue signal sent from the DUT.
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 with the port specified in the received an About announcement.
* The test device calls the Execute()method on the DUT’s centralized management object with isReturn, deviceId, objectPath, interfaceName, methodName(i.e., ApplianceContol() method) and method arguments maintained with the validation test.
* The test device receives the method call from the DUT to call the applianceControl() method of the test device with interface name, method name, and method arguments that is provided in the Execute() method call.
* The test device sends the replied value of the applianceControl() method call to the DUT.
* The test device receives the replied value included in the ReturnValue signal sent from the DUT.

## Smart home-v1-04: ApplianceUnRegistration() method call

**Objective**

Verify that the test device can call the ApplianceUnRegistration() method to leave the centralized management system.

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 with the port specified in the received About announcement.
3. The test device calls the ApplianceRegistration() method on the DUT’s centralized management object with the registration parameters.
4. The test device calls the Verification() method on the DUT’s centralized management object with device Id to verify whether the test device registers on the DUT successfully.
5. The test device waits to receive the returned parameters (i.e., well known name, unique name and device Id) according to the device Id from the DUT.
6. The test device calls the ApplianceUnRegistration() method on the DUT’s centralized management object with device Id to unregister the test device from the DUT.
7. The test device calls the Verification() method on the DUT’s centralized management object with device Id to verify whether the test device unregisters the test device from the DUT successfully.
8. The test device waits to receive the reply from the DUT.
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 with the port specified in the received About announcement.
* The test device calls the ApplianceRegistration() method on the DUT’s centralized management object to provide the registration parameters (i.e., well known name, unique name and device Id) and the method returns a OK response.
* The test device calls the Verification() method on the DUT’s centralized management object with device Id.
* The received parameters returned from the Verification() method on the DUT’s centralized management object is the same as the test device’s parameters (i.e., well known name , unique name and device Id) according to the device Id provided by the test device.
* The test device calls the ApplianceUnRegistration() method on the DUT’s centralized management object with device Id and the method returns a OK response.
* The test device calls the Verification() method on the DUT’s centralized management object with device Id.
* The received parameters returned from the Verification() method on the DUT’s centralized management object is NULL.

## Smart home-v1-05: DeviceHeartBeat () method call

Objective

Verify that the test device can call the DeviceHeartBeat () method to keep the device present and alive in the centralized management system.

Note that in the Smart Home service, the DeviceHeartBeat() needs to be called at regular intervals i.e., 10s to keep the device present and alive.

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 with the port specified in the received About announcement.
3. The test device calls the ApplianceRegistration() method on the DUT’s centralized management object with the registration parameters.
4. The test device calls the Verification() method on the DUT’s centralized management object with device Id to verify whether the test device registers on the DUT successfully.
5. The test device waits to receive the returned parameters (i.e., well known name, unique name, device Id) according to the device Id from the DUT.
6. The test device calls the DeviceHeartBeat () method on the DUT’s centralized management object with device Id and ValidateCode at regular intervals (every 10s).
7. The test device waits to receive the ReturnValue signal sent from the DUT.
8. The test device calls the Verification() method on the DUT’s centralized management object with device Id to verify whether the test device keeps alive in the centralized management system.
9. The test device waits to receive the returned parameters (i.e., well known name, unique name and device Id) according to the device Id from the DUT.
10. The test device doesn’t call the DeviceHeartBeat () method in 30s.
11. The test device calls the Verification() method on the DUT’s centralized management object with device Id to verify whether the test device keeps alive in the centralized management system.
12. The test device waits to receive the reply from the DUT.
13. 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 with the port specified in the received About announcement.
* The test device calls the ApplianceRegistration() method on the DUT’s centralized management object to provide the registration parameters (i.e., well known name, unique name, device Id) and the method returns a OK response.
* The test device calls the Verification() method on the DUT’s centralized management object with device Id.
* The received parameters returned from the Verification() method on the DUT’s centralized management object is the same as the test device’s parameters (i.e., well known name , unique name, device Id) according to the device Id provided by the test device.
* The test device calls the DeviceHeartBeat() method on the DUT’s centralized management object with device Id and ValidateCode at regular intervals (every 10s).
* The test device receives the status included in the ReturnValue signal sent from the DUT.
* The test device calls the Verification() method on the DUT’s centralized management object with device Id.
* The received parameters returned from the Verification() method on the DUT’s centralized management object include well known name , unique name and device Id according to the device Id provided by the test device.
* The test device calls the Verification() method on the DUT’s centralized management object with device Id.
* The received parameters returned from the Verification() method on the DUT’s centralized management object is NULL.