ALLSEEN ALLIANCE

AllJoyn™ Notification Service Framework Test Case Specifications

March 21, 2014

Contents

1 Introduction

1.1 Purpose

These test cases evaluate and verify the functionality related to the AllJoyn Notification service framework when used by an application to do one or both of the following using the Notification interface:

- Receive (or consume) notifications from other applications
- Send (or produce) notifications to other applications

The Notification interface is used by an AllJoyn application to send events or state update notifications to other devices connected to an end user's home network, such as a Wi-Fi network.

1.2 Scope

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

1.3 References

The following are reference documents.

- *AllJoyn™ Notification Service Framework 1.0 Interface Specification*
- *AllJoyn™ About Feature 1.0 Interface Specification*

2 Environment setup

2.1 Requirements

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

- An AllJoyn-enabled device (the device under test or DUT) that supports the AllJoyn Notification service framework 1.0 in the role of a consumer, producer, or both.
- A supported test device on which the test cases will run
- A Wi-Fi access point (referred to as the personal AP).

2.2 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
- If the DUT supports the Notification service in the role of a consumer, at least one process on the DUT is acting as a Notification Consumer and listening for notifications through the Notification interface
- If the DUT supports the Notification service in the role of a producer, at least one process on the DUT is announcing its capabilities through its About announcement, including its support for the Notification interface.

2.3 Parameters

Table 1. Parameters for the Notification service framework

Parameter	Description
DeviceId	Device ID of the DUT
AppId	Application ID of the Notification Consumer or Notification Producer application on the DUT

3 Notification service framework test cases

3.1 Notification-Consumer-v1-01: Basic text message

Objective

Verify the DUT receives and displays a basic text message that supports the English language.

Procedure

1. The test device registers an About bus object on its bus and sends an About announcement.
2. The test device registers Notification bus objects on its bus (one for each message type).
3. The test device randomly picks from one of the following notifications texts:
   - Test Msg 1
   - Test Msg 2
   - Test Msg 3
4. The test device sends a notify signal from a Notification bus object (the one for the Information message type) with a TTL of 2 minutes and parameters set to values as follows:
   - version: 1
   - msgType: 2 (Information message)
   - List<langText>: one langText element with a langTag of “en” and the text set to the notification text picked above.
   - deviceId, deviceName, appId, and appName set according to what has been specified in the About bus object.
5. The test device prompts the tester to select the text of the notification received based on the notification displayed on the DUT.
6. The test device leaves the session.

Expected results

- The DUT receives the notify session-less signal and displays the text of the notification as an information message.
- The tester responds to the prompt by selecting the correct notification text based on the notification text displayed on the DUT.

3.2 Notification-Consumer-v1-02: Multiple languages

Objective

Verify the DUT receives and displays a text message containing multiple languages.

Procedure

1. The test device registers an About bus object on its bus and sends an About announcement.
2. The test device registers Notification bus objects on its bus (one for each message type).
3. The test device randomly picks from one of the following notifications texts:
   - Two languages Msg 1
   - Two languages Msg 2
   - Two languages Msg 3
4. The test device sends a notify signal from a Notification bus object (the one for the Information message type) with a TTL of 2 minutes and parameters set to values as follows:
   - version: 1
   - msgType: 2 (Information message)
   - List<langText>: two langText elements, one with a langTag of “en” and one with a langTag of “fr” and both with the text set to the notification text picked above
   - deviceId, deviceName, appId, and appName set according to what has been specified in the About bus object
5. The test device prompts the tester to select the text of the notification received based on the notification displayed on the DUT.
6. The test device leaves the session.

Expected results

- The DUT receives the notify session-less signal and displays at least one of the language variants as an information message.
- The tester responds to the prompt by selecting the correct notification text based on the notification text displayed on the DUT.

3.3 Notification-Consumer-v1-04: Invalid language field

Objective

Verify the DUT receives and displays a text message with an invalid language field.

Procedure

1. The test device registers an About bus object on its bus and sends an About announcement.
2. The test device registers Notification bus objects on its bus (one for each message type).
3. The test device randomly picks from one of the following notifications texts:
   - Invalid langTag Msg 1
   - Invalid langTag Msg 2
   - Invalid langTag Msg 3
4. The test device sends a notify signal from a Notification bus object (the one for the Information message type) with a TTL of 2 minutes and parameters set to values as follows:
   - version: 1
   - msgType: 2 (Information message)
   - List<langText>: one langText element with a langTag of “INVALID” and the text set to the notification text picked above
   - deviceId, deviceName, appId, and appName set according to what has been specified in the About bus object
5. The test device prompts the tester to select the text of the notification received based on the notification displayed on the DUT.
6. The test device leaves the session.

Expected results

- The DUT receives the notify session-less signal and displays the text of the notification as an information message.
- The tester responds to the prompt by selecting the correct notification text based on the notification text displayed on the DUT.

3.4 Notification-Consumer-v1-05: Message priority

Objective

Verify the DUT receives and displays text messages with the appropriate message priorities.

Procedure

1. The test device registers an About bus object on its bus and sends an About announcement.
2. The test device registers Notification bus objects on its bus (one for each message type).

3. The test device randomly picks from one of the following sets of notification texts:
   - Priority Msg 1 (Emergency); Priority Msg 2 (Warning); Priority Msg 3 (Information)
   - Priority Msg 1 (Emergency); Priority Msg 2 (Information); Priority Msg 3 (Warning)
   - Priority Msg 1 (Warning); Priority Msg 2 (Emergency); Priority Msg 3 (Information)
   - Priority Msg 1 (Warning); Priority Msg 2 (Information); Priority Msg 3 (Emergency)
   - Priority Msg 1 (Information); Priority Msg 2 (Emergency); Priority Msg 3 (Warning)
   - Priority Msg 1 (Information); Priority Msg 2 (Warning); Priority Msg 3 (Emergency)
4. The test device sends notify signals from each Notification bus object (one for each message type) with a TTL of 2 minutes and parameters set to values as follows:
   - version: 1
   - msgType: 0 – Emergency, 1 – Warning, or 2 – Information
   - List<langText>: one langText element with a langTag of “en” and the text set to the notification text picked above for the given msgType
   - deviceId, deviceName, appId, and appName set according to what has been specified in the About bus object
5. The test device prompts the tester to select the set of notification texts received based on the notifications displayed on the DUT.
6. The test device leaves the session.

Expected results

- The DUT receives the notify session-less signals and displays the text of the notifications as the appropriate message type.
- The tester responds to the prompt by selecting the correct set of notification texts based on the notification texts displayed on the DUT.

3.5 Notification-Consumer-v1-06: Custom attributes

Objective

Verify the DUT receives and displays a text message containing custom attributes.

Procedure

1. The test device registers an About bus object on its bus and sends an About announcement.
2. The test device registers Notification bus objects on its bus (one for each message type).
3. The test device randomly picks from one of the following notifications texts:
   - Msg w/ attributes 1
   - Msg w/ attributes 2

- Msg w/ attributes 3

4. The test device sends a notify signal from a Notification bus object (the one for the Information message type) with a TTL of 2 minutes and parameters set to values as follows:
   - version: 1
   - msgType: 2 (Information message)
   - List<langText>: one langText element with a langTag of “en” and the text set to the notification text picked above
   - deviceId, deviceName, appId, and appName set according to what has been specified in the About bus object
   - List<attribute>: one attribute element with an attrName of “org.alljoyn.validation.test” and attrValue of “value”
5. The test device prompts the tester to select the text of the notification received based on the notification displayed on the DUT.
6. The test device leaves the session.

Expected results

- The DUT receives the notify session-less signal and displays the text of the notification as an information message.
- The tester responds to the prompt by selecting the correct notification text based on the notification text displayed on the DUT.

4 Notification Producer test cases

4.1 Notification-v1-01: Sending of notifications

Objective

Verify the DUT sends notifications according to the Notification interface specification.

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 prompts the tester to respond when all the notifications have been sent from the DUT.
4. Until the tester responds, the test device listens for notifications and validates all notifications received which match the deviceId and appId of the application.
5. Once the tester responds, the test device leaves the session.

Expected results

- At least one notification is received before the test ends.
- For each notification received:
  - The version field is 1.
  - The deviceId, deviceName, appId, and appName fields are present and match the values in the received About announcement.
  - The msgType field is 0, 1, or 2.
  - If a richIconUrl attribute (attrName = 0) is present then the iconUrl in the attrValue is a valid URL.
  - If a richAudioUrl attribute (attrName = 1) is present then each audioUrl in the attrValue is a valid URL.
  - If a respObjectPath attribute (attrName = 4) is present then the attrValue is a valid path and an object can be found at this path on the application’s message bug