AllJoyn™ Core Software Test Case Specifications

Version 14.06 Update 1

October 21, 2014

Contents

1 Introduction

1.1 Purpose

These test cases evaluate and verify the functionality related to the AllJoyn™ Core software exposed by a device through the relevant AllJoyn interfaces such as the About feature.

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.

The Introspectable interface makes the Events and Actions feature discoverable through the introspected BusObject. The introspection XML provides description elements under Objects, Interfaces, Methods (including parameters), Signals (including parameters), and Properties. These descriptions are used to define the Events and Actions rules.

NOTE

Refer to the appropriate *AllJoyn™ Interface Definition* document to understand the AllSeen Alliance's Compliance and Certification requirements and how to use this document to help ensure that your application meets those requirements.

1.2 Scope

These test cases are designed to determine if a device conforms to the Core interfaces. For a device to conform, these test cases must be executed against each application on the device announcing its capabilities through an About announcement, as well as other feature specifications included in the AllJoyn Core software. Successful completion of all test cases in this document does not guarantee that the tested device will interoperate with other devices.

1.3 Release history

Release version	Date	What changed in this document
14.02	March 21, 2014	Initial release as the *AllJoyn™ About Feature 1.0 Test Case Specifications* document.
14.06	June 30, 2014	■ Renamed the *AllJoyn™ Core Software 14.06 Test Case Specifications* document. ■ Added release history section ■ Added a note to address mandatory information in the appropriate *Interface Definition* document.
14.06 Update 1	October 21, 2014	Added the Introspectable Test Cases chapter to support the Events and Actions feature.

1.4 References

The following are reference documents.

- *AllJoyn™ About Interface Definition*
- RFC 2396 (Uniform Resource Identifiers (URI): Generic Syntax)
- *Guide to Using the AllJoyn™ Events and Actions Feature*

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 About feature 1.0 and implements the Events and Actions feature.
- 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
- At least one application on the DUT is announcing its capabilities through an About announcement.
- The DUT is announcing its support of the Introspectable interface through its About announcement.

2.3 Parameters

Table 1. Parameters for the AllJoyn core

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

3 About Test Cases

3.1 About-v1-01: About announcement received

Objective

Verify that the test device receives a valid About announcement from the application on the DUT and that it can join a session with the application at the port specified in the About announcement.

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 verifies the received About announcement's parameters.
3. The test device joins a session with the application at the port specified in the received About announcement.
4. The test device leaves the session.

Expected results

- The test device receives an About announcement from the application on the DUT.
- The received About announcement's version parameter is 1.
- The received About announcement's objectDescription parameter contains an element with a path of "/About" which includes the org.alljoyn.About interface
- The received About announcement's metaData parameter contains the following required fields:
  - AppId
  - DefaultLanguage
  - DeviceName
  - DeviceId
  - AppName
  - Manufacturer
  - ModelNumber
- The test device joins a session with the application at the port specified in the received About announcement.

3.2 About-v1-02: Version property consistent with the About announcement

Objective

Verify that the Version property retrieved from the application's About bus object matches the version parameter in its About announcement.

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 retrieves the Version property from the application's About bus object and verifies that it matches the version parameter in the received About announcement.
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 Version property retrieved from the application's About bus object matches the version parameter in the received About announcement.

3.3 About-v1-03: GetObjectDescription() consistent with the About announcement

Objective

Verify that the set of paths and interfaces retrieved from the GetObjectDescription() method on the application's About bus object is the same as the set of paths and interfaces in the objectDescription parameter received in its About announcement.

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 GetObjectDescription() method on the application's About bus object and verifies that the returned set of paths and interfaces is the same as those in the objectDescription parameter in the received About announcement.

4. The test device leaves the session with the application.

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 set of paths and interfaces returned from the GetObjectDescription() method on the application's About bus object is the same as the set of paths and interfaces in the objectDescription parameter in the received About announcement.

3.4 About-v1-04: Bus objects consistent with the About announcement

Objective

Verify that the application's set of bus objects and their interfaces includes the set of paths and interfaces in the objectDescription parameter received in its About announcement.

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 uses the Introspect() method on the Introspectable interface to introspect the application's message bus and verify that the set of bus objects and their interfaces includes the set of paths and interfaces in the objectDescription parameter in the received About announcement.
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 test device verifies through introspection that the set of bus objects and their interfaces includes the set of paths and interfaces in the objectDescription parameter in the received About announcement.
- The test device verifies the values match what is present in the About announcement.

3.5 About-v1-05: Standardized interfaces match definitions

Objective

Verify that definitions for all "org.alljoyn" interfaces supported by objects on the application's message bus are valid and match the standardized definitions.

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 uses the Introspect() method on the Introspectable interface to introspect the application's message bus and retrieve the set of objects and their interfaces.
4. The test device verifies that the definitions for all "org.alljoyn" interfaces supported by objects on the application's message bus are valid and match the standardized definitions.
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.
- The test device retrieves the set of objects and their interfaces existing on the application's message bus using the Introspect() method on the Introspectable interface.
- The definitions for all "org.alljoyn" interfaces supported by objects on the application's message bus are valid and match the standardized definitions.

3.6 About-v1-06: GetAboutData() with default language

Objective

Verify the metadata fields returned from a call to the GetAboutData() method with the languageTag parameter set to the default language.

The metadata fields returned must include all of the required fields and the values of any fields defined in the Interface definition must match their defined signature and any defined format.

The defined fields received in the metaData parameter of the application's About announcement must be included in the metadata fields returned and the values must be the same.

Procedure

1. The test device waits to receive 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 GetAboutData() method on the application's About bus object with the languageTag parameter set to the value of the DefaultLanguage field in the metaData parameter of the received About announcement.
4. The test device verifies that the returned metadata fields include all the required fields.
5. The test device verifies that the values for all of the defined fields in the returned metadata fields match their defined signatures.
6. The test device verifies that the returned metadata fields include all the defined fields in the metaData parameter of the received About announcement and their values are the same.
7. The test device verifies that if the returned metadata fields include the DateOfManufacture field then the value matches the defined format ("YYYY-MM-DD") and it represents a valid date.
8. The test device verifies that if the returned metadata fields include the SupportUrl field then the value is a properly formatted URL (as defined by RFC 2396).
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.
- The GetAboutData() method call with the languageTag parameter set to the default language returns a set of metadata fields.
- The set of returned metadata fields includes the following required fields:
  - AppId
  - DefaultLanguage
  - DeviceName
  - DeviceId
  - AppName
  - Manufacturer
  - ModelNumber
  - SupportedLanguages
  - Description

  - SoftwareVersion
  - AJSoftwareVersion
- The values for all of the defined fields in the returned metadata fields match their defined signatures:
  - AppId: ay
  - DefaultLanguage: s
  - DeviceName: s
  - DeviceId: s
  - AppName: s
  - Manufacturer: s
  - ModelNumber: s
  - SupportedLanguages: as
  - Description: s
  - DateOfManufacture: s
  - SoftwareVersion: s
  - AJSoftwareVersion: s
  - HardwareVersion: s
  - SupportUrl: s
- The returned metadata fields include all the defined fields in the metaData parameter of the received About announcement and their values are the same.
- If the returned metadata fields include the DateOfManufacture field then the value matches the defined format (“YYYY-MM-DD”) and it represents a valid date.
- If the returned metadata fields include the SupportUrl field then the value is a properly formatted URL (as defined by RFC 2396).

3.7 About-v1-07: GetAboutData() with each supported language

Objective

Verify the metadata fields returned from calls to the GetAboutData() method with the languageTag parameter set to each of the supported languages.

The metadata fields returned from each call to the GetAboutData() method must include all of the required fields and the values of any fields defined in the interface definition must match their defined signature and any defined format.

The value received for any field defined as non-localized must match the value received for the default language.

Procedure

1. The test device waits to receive 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 GetAboutData() method on the application's About bus object with the languageTag parameter set to the value of the DefaultLanguage field in the metaData parameter of the received About announcement.
4. The test device retrieves the list of supported languages from the SupportedLanguages field in the returned metadata fields.
5. If the list of supported languages contains only one language then the test case exits with a note to this effect.
6. The test device calls the GetAboutData() method on the application's About bus object for each of the supported languages (except for the default language), providing each language as the languageTag parameter.
7. The test device verifies that the returned metadata fields from each call include all the required fields.
8. The test device verifies that the value received for any field defined as non-localized is the same for all of the supported languages.
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.
- The GetAboutData() method call with the languageTag parameter set to the default language returns a set of metadata fields.
- If the list of supported languages contains more than one language then the GetAboutData() method calls with the languageTag parameter set to each of the other supported languages return a set of metadata fields.
- The set of returned metadata fields for each language includes the following required fields:
  - AppId
  - DefaultLanguage
  - DeviceName
  - DeviceId
  - AppName
  - Manufacturer

  - ModelNumber
  - SupportedLanguages
  - Description
  - SoftwareVersion
  - AJSoftwareVersion
- The value received for any field defined as non-localized (listed below) is the same for all of the supported languages.
  - AppId
  - DefaultLanguage
  - DeviceId
  - ModelNumber
  - SupportedLanguages
  - DateOfManufacture
  - SoftwareVersion
  - AJSoftwareVersion
  - HardwareVersion
  - SupportUrl

3.8 About-v1-08: GetAboutData() without a specified language

Objective

Verify the metadata fields returned from a call to the GetAboutData() method with the languageTag parameter set to a zero length string.

The set of defined fields in the metadata fields returned must be the same set of fields with the same values as returned when the languageTag parameter is set to the default language.

Procedure

1. The test device waits to receive 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 GetAboutData() method on the application's About bus object with the languageTag parameter set to the value of the DefaultLanguage field in the metaData parameter of the received About announcement.

4. The test device calls the GetAboutData() method on the application's About bus object with the languageTag parameter set to a zero length string.
5. The test device verifies that the set of defined fields in the metadata fields returned are the same set of fields with the same values between the two GetAboutData() method calls.
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.
- The GetAboutData() method call with the languageTag parameter set to the default language returns a set of metadata fields.
- The GetAboutData() method call with the languageTag parameter set to a zero length string returns a set of metadata fields.
- The set of defined fields in the returned metadata fields are the same set of fields with the same values between the two GetAboutData() method calls

3.9 About-v1-09: GetAboutData() for an unsupported language

Objective

Verify that the GetAboutData() method returns the error "org.alljoyn.Error.LanguageNotSupported" when the languageTag parameter is set to an unsupported language.

Procedure

1. The test device waits to receive 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 GetAboutData() method on the application's About object with the languageTag parameter set to "INVALID" (an unsupported language).
4. The test device verifies the method call returned the error "org.alljoyn.Error.LanguageNotSupported".
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.
- The GetAboutData() method call with the languageTag parameter set to “INVALID” returns the error “org.alljoyn.Error.LanguageNotSupported”.

3.10 About-v1-10: GetContent() on the About DeviceIcon

Objective

Verify that the MimeType and Size Properties and the GetContent() method call on the About DeviceIcon object (if supported by the application).

Procedure

1. The test device waits to receive 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. If no object exists at the path “/About/DeviceIcon” implementing the org.alljoyn.Icon interface then the test case exits with a note that the application does not support the About DeviceIcon.
4. The test device retrieves the value of the MimeType and Size properties from the About DeviceIcon object.
5. If the MimeType value is a zero length string and the Size value is zero then the test case exits with a note that the application does not provide an icon through the GetContent() method.
6. The test device verifies that the MimeType value matches the regular expression “image/*”.
7. The test device verifies that the Size value is less than ALLJOYN_MAX_ARRAY_LEN (131072 bytes).
8. The test device calls the GetContent() method on the About DeviceIcon object to retrieve the binary image data.
9. The test device verifies that the Size value equals the size of the byte array returned from the GetContent() method.
10. The test device verifies that the binary image data can be loaded into memory using the Android BitmapFactory.decodeByteArray() method.
11. 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.

- If no object exists at the path “About/DeviceIcon” implementing the org.alljoyn.Icon interface then the test case exits with a note to that effect.
- The test device retrieves the value of the MimeType and Size properties from the About DeviceIcon object.
- If the MimeType value is a zero length string and the Size value is zero then the test case exits with a note that the application does not provide an icon through the GetContent() method.
- The MimeType value matches the regular expression “image/*”.
- The Size value is less than ALLJOYN_MAX_ARRAY_LEN (131072 bytes).
- The test device receives the binary image data from the GetContent() method call.
- The Size value matches the size of the binary image data.
- The binary image data is loaded into memory using the Android BitmapFactory.decodeByteArray() method.

3.11 About-v1-11: GetUrl() on the About DeviceIcon

Objective

Verify that the GetUrl() method call on the About DeviceIcon object (if supported by the application) returns a valid URL.

Procedure

1. The test device waits to receive 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. If no object exists at the path “/About/DeviceIcon” implementing the org.alljoyn.Icon interface then the test case exits with a note that the application does not support the About DeviceIcon.
4. The test device calls the GetUrl() method on the Icon interface.
5. If the value returned from the GetUrl() method call is a zero length string then the test case exits with a note that the application does not provide an icon through the GetUrl() method.
6. The test device verifies that the value is a properly formatted URL.
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.

- If no object exists at the path “About/DeviceIcon” implementing the org.alljoyn.Icon interface then the test case exits with a note to that effect.
- The test device calls the GetUrl() method on the Icon interface.
- If the value returned from the GetUrl() method is a zero length string then the test case exits with a note that the application does not provide an icon through the GetUrl() method.
- The value returned from the GetUrl() method is a properly formatted URL.

4 Introspectable Test Cases

4.1 EventsActions-v1-01: Description tag existence in introspection XML and identical XML across different description languages

Objective

Verify that introspection XML of a Bus Object that announces itself as implementing the Introspectable interface (org.allseen.Introspectable) contains a description tag.

If the introspection XML of the Bus Object doesn't contain a description tag, one of its child objects should contain it.

If a Bus Object has a description in more than one language, the introspection XML retrieved for each language should be identical. (This is to ensure that descriptions were provided for each object with description in all languages.)

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. For each matching bus object found, the test device performs the following steps:
    a. Call the GetDescriptionLanguages() method of the Introspectable interface.
    b. Call IntrospectWithDescription() for each of the description languages.
    c. Remove the content of the description tag from the received introspection XML. Hold the XML for the later use for the first language in the list of description languages.
    d. Verify for all of the subsequent languages that the introspection XML is identical to the XML of the first language.
    e. Introspect the child objects of the currently tested Bus Object and repeat this step.
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 announcement.

- For each bus object implementing the Introspectable interface:
  - Retrieve an array of the description languages supported by the introspected bus object. If the array is empty, it means that the tested object has no description tag, so the child objects are introspected.
  - Retrieve the introspection XML with description tags in the requested language.
  - The content of the description tags should be removed, so the tags receive the form of: <description></description>.
  - The introspection XMLs are identical for each of the description languages by comparing it to the XML of the first language.
  - Retrieve the introspection XML of the child objects of the tested bus object and repeat step 3 on each of them.

The test fails if no bus object with the description tag was found or if a difference was found between the introspection XML of the different description languages.