ALLSEEN ALLIANCE

AllJoyn™ Control Panel Service Framework 14.06 Test Case Specifications

June 30, 2014

Contents

1 Introduction

1.1 Purpose

These test cases evaluate and verify the functionality related to the AllJoyn Control Panel service framework collection of interfaces.

These interfaces provide the Control Panel service framework a mechanism to allow for a controller application to render the UI based on the controllee application’s widget metadata.

NOTE

See Interface Data to review the parameters supported by the Control Panel service framework. Refer to the *AllJoyn™ Control Panel Service Framework Interface Specification* for more information (description, signature, etc.).

1.2 Scope

These test cases are designed to determine if a device conforms to the Control Panel 1.0 interface specifications. Successful completion of all test cases does not guarantee that the tested device will interoperate with other devices.

1.3 Release history

Release version	What changed in this document
14.02	Initial release.
14.06	■ Added release history section ■ Added a note to address mandatory data in the appropriate *Interface Specification* document. ■ Added a chapter to list the mandatory/optional interface data

1.4 References

The following are reference documents

- *AllJoyn™ Control Panel Service Framework 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 implements the Control Panel interfaces according to the Control Panel service framework 1/0 specification.
- 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 process on the DUT is announcing its capabilities through its About announcement, including its support for the ControlPanel interface
- The passcode for the DUT is set to the default passcode.

2.3 Parameters

Table 1. Parameters for the Control Panel service framework

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

3 Control Panel Service Framework Test Cases

3.1 ControlPanel-v1-01: Verify all ControlPanel bus objects

Objective

Verify that all bus objects implementing the ControlPanel interface conform to the specifications.

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 registers an AuthListener on the AllJoyn framework that provides the default passcode ("000000") when authentication is requested.
4. The test device uses the Introspect() method on the Introspectable interface to locate all bus objects implementing the org.alljoyn.ControlPanel.ControlPanel interface.
5. For each matching bus object found, the test device performs the following steps:
   a. Retrieve the Version property.
   b. Use the Introspect() method on the Introspectable interface to locate child bus objects which implement the org.alljoyn.ControlPanel.Container or org.alljoyn.ControlPanel.SecuredContainer interface.
6. The test device leaves the session.

Expected results

- If no bus objects implementing the ControlPanel interface are found, a note is added to this effect.
- For each bus object implementing the ControlPanel interface:
  - The object's path matches the pattern "/ControlPanel/[^/]+/[^/]+
  - The Version property is 1
  - At least one child bus object implementing the org.alljoyn.ControlPanel.Container or org.alljoyn.ControlPanel.SecuredContainer interface exists.

3.2 ControlPanel-v1-02: Verify all Container bus objects

Objective

Verify that all bus objects implementing the Container or SecuredContainer interface conform to the specifications.

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 registers an AuthListener in the AllJoyn framework that provides the default passcode ("000000") when authentication is requested.
4. The test device uses the Introspect() method on the Introspectable interface to locate all bus objects implementing the org.alljoyn.ControlPanel.Container or org.alljoyn.ControlPanel.SecuredContainer interface.
5. For each matching bus object found, the test device performs the following steps:
   a. Retrieve the Version property.
   b. Retrieve the States property.
   c. Retrieve the OptParams property.
6. The test device leaves the session.

Expected results

- If no bus objects implementing the Container or SecuredContainer interface are found, a note is added to this effect.
- For each bus object implementing the Container or SecuredContainer interface:
  - The bus object is a descendant of a bus object implementing one of the following interfaces:
    - org.alljoyn.ControlPanel.ControlPanel
    - org.alljoyn.ControlPanel.NotificationAction
    - org.alljoyn.ControlPanel.ListProperty
    - org.alljoyn.ControlPanel.SecuredListProperty
    - org.alljoyn.ControlPanel.Container
    - org.alljoyn.ControlPanel.SecuredContainer
  - The Version property is 1
  - The States property is 0x00 or 0x01
  - The OptParams property is set as follows:
    - If OptParams contains a key with '0', the value type is a string.

  - If OptParams contains a key with '1', the value type is a uint32.
  - If OptParams contains a key with '2', the value type is an array of uint16, the array has a size of one, and the value of the one element must be one of the following:
    - 1 (Vertical Linear)
    - 2 (Horizontal Linear)

3.3 ControlPanel-v1-03: Verify all Property bus objects

Objective

Verify that all bus objects implementing the Property or SecuredProperty interface conform to the specifications.

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 registers an AuthListener in the AllJoyn framework that provides the default passcode ("000000") when authentication is requested.
4. The test device uses the Introspect() method on the Introspectable interface to locate all bus objects implementing the org.alljoyn.ControlPanel.Property or org.alljoyn.ControlPanel.SecuredProperty interface.
5. For each matching bus object found, the test device performs the following steps:
   a. Retrieve the Version property.
   b. Retrieve the States property.
   c. Retrieve the OptParams property.
   d. Retrieve the Value property.
   e. Use the Introspect() method on the Introspectable interface to determine its parent bus object.
6. The test device leaves the session.

Expected results

- If no bus objects implementing the Property or SecuredProperty interface are found, a note is added to this effect.
- For each bus object implementing the Property or SecuredProperty interface:
  - The Version property is 1
  - The States property is 0x00, 0x01, or 0x02
  - The OptParams property is set as follows:

- If OptParams contains a key with '0', the value type is a string.
- If OptParams contains a key with '1', the value type is a uint32.
- If OptParams contains a key with '2' (the hints field), the value type is an array of uint16, the array has a size of one, and the value of the one element must be one of the following:
  - 1 (Switch)
  - 2 (CheckBox)
  - 3 (Spinner)
  - 4 (RadioButton)
  - 5 (Slider)
  - 6 (TimePicker)
  - 7 (DatePicker)
  - 8 (NumberPicker)
  - 9 (NumericKeypad)
  - 10 (RotaryKnob)
  - 11 (TextLabel)
  - 12 (NumericView)
  - 13 (EditText)
- If OptParams contains a key with '3', the value type is a string.
- If OptParams contains a key with '4', the value type is "a(vs)" where all variant instances match the type of the Value property.
- If OptParams contains a key with '5', the value type is "vvv" where the type of each variant matches the type of the Value property (representing the min, max, and increment for the Value) and min ≤ max.
- If the hint is 1 (Switch), the type of the Value property is boolean.
- If the hint is 2 (CheckBox), 3 (Spinner), or 4 (RadioButton), OptParams must contain a key with '4' with a list of one or more values.
- If the hint is 5 (Slider) or 10 (RotaryKnob), the type of the Value property is numeric.
- If the hint is 6 (TimePicker), the type of the Value property is a composite type of q(qqq) where the first value (representing the composite type) is 1.
- If the hint is 7 (DatePicker), the type of the Value property is a composite type of q(qqq) where the first value (representing the composite type) is 0.
- If the hint is 8 (NumberPicker), the type of the Value property is numeric.
- If the hint is 9 (NumericKeypad), the type of the Value property is numeric.
- If the hint is 11 (TextLabel), the type of the Value property is a string.

  - If the hint is 12 (NumericView), the type of the Value property is numeric.
  - If the hint is 13 (EditText), the type of the Value property is a string.
- The bus object is a descendant of a bus object implementing one of the following interfaces:
  - org.alljoyn.ControlPanel.Container
  - org.alljoyn.ControlPanel.SecuredContainer

3.4 ControlPanel-v1-04: Verify all LabelProperty bus objects

Objective

Verify that all bus objects implementing the LabelProperty interface conform to the specifications.

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 registers an AuthListener with the AllJoyn framework that provides the default passcode ("000000") when authentication is requested.
4. The test device uses the Introspect() method on the Introspectable interface to locate all bus objects implementing the org.alljoyn.ControlPanel.LabelProperty interface.
5. For each matching bus object found, the test device performs the following steps:
   a. Retrieve the Version property.
   b. Retrieve the States property.
   c. Retrieve the Label property.
   d. Retrieve the OptParams property.
6. The test device leaves the session.

Expected results

- If no bus objects implementing the LabelProperty interface are found, a note is added to this effect.
- For each bus object implementing the LabelProperty interface:
  - The Version property is 1
  - The States property is 0x00 or 0x01
  - The OptParams property is set as follows:
    - If OptParams contains a key with '1', the value type is a uint32.

    - If OptParams contains a key with '2' (the hints field), the value type is an array of uint16, the array has a size of one, and the value of the one element must be 1 (TextLabel).
  - ❐ The bus object is a descendant of a bus object implementing one of the following interfaces:
    - org.alljoyn.ControlPanel.Container
    - org.alljoyn.ControlPanel.SecuredContainer

3.5 ControlPanel-v1-05: Verify all Action bus objects

Objective

Verify that all bus objects implementing the Action or SecuredAction interface conform to the specifications.

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 registers an AuthListener with the AllJoyn framework that provides the default passcode ("000000") when authentication is requested.
4. The test device uses the Introspect() method on the Introspectable interface to locate all bus objects implementing the org.alljoyn.ControlPanel.Action or org.alljoyn.ControlPanel.SecuredAction interface.
5. For each matching bus object found, the test device performs the following steps:
   a. Retrieve the Version property.
   b. Retrieve the States property.
   c. Retrieve the OptParams property.
6. The test device leaves the session.

Expected results

- ■ If no bus objects implementing the Action or SecuredAction interface are found, a note is added to this effect.
- ■ For each bus object implementing the Action or SecuredAction interface:
  - ❐ The Version property is 1
  - ❐ The States property is 0x00 or 0x01
  - ❐ The OptParams property is set as follows:
    - If OptParams contains a key with '0', the value type is a string.
    - If OptParams contains a key with '1', the value type is a uint32.

    - If OptParams contains a key with '2' (the hints field), the value type is an array of uint16, the array has a size of one, and the value of the one element must be 1 (ActionButton).
  - ❐ The bus object is a descendant of a bus object implementing one of the following interfaces:
    - org.alljoyn.ControlPanel.Container
    - org.alljoyn.ControlPanel.SecuredContainer

3.6 ControlPanel-v1-06: Verify all Dialog bus objects

Objective

Verify that all bus objects implementing the Dialog or SecuredDialog interface conform to the specifications.

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 registers an AuthListener with the AllJoyn framework that provides the default passcode ("000000") when authentication is requested.
4. The test device uses the Introspect() method on the Introspectable interface to locate all bus objects implementing the org.alljoyn.ControlPanel.Dialog or org.alljoyn.ControlPanel.SecuredDialog interface.
5. For each matching bus object found, the test device performs the following steps:
   a. Retrieve the Version property.
   b. Retrieve the States property.
   c. Retrieve the OptParams property.
   d. Retrieve the Message property.
   e. Retrieve the NumActions property.
      - If NumActions = 2, call the method Action3.
      - If NumActions = 1, call the methods Action2 and Action3.
6. The test device leaves the session.

Expected results

- ■ If no bus objects implementing the Dialog or SecuredDialog interface are found, a note is added to this effect.
- ■ For each bus object implementing the Dialog or SecuredDialog interface:
  - ❐ The Version property is 1

- ❐ The States property is 0x00 or 0x01
- ❐ The OptParams property is set as follows:
  - If OptParams contains a key with '0', the value type is a string.
  - If OptParams contains a key with '1', the value type is a uint32.
  - If OptParams contains a key with '2' (the hints field), the value type is an array of uint16, the array has a size of one, and the value of the one element must be 1 (AlertDialog).
  - If OptParams contains a key with '6', the value type is a string.
  - If the NumActions Property is ≥ 2, OptParams contains a key with '7' and the value type is a string.
  - If the NumActions Property is equal to 3, OptParams contains a key with '8' and the value type is a string.
- ❐ All calls made to the methods, Action2 and Action3, return a MethodNotAllowed error.
- ❐ The bus object is a descendant of a bus object implementing one of the following interfaces:
  - org.alljoyn.ControlPanel.NotificationAction
  - org.alljoyn.ControlPanel.Action
  - org.alljoyn.ControlPanel.SecuredAction

3.7 ControlPanel-v1-07: Verify all ListProperty bus objects

Objective

Verify that all bus objects implementing the ListProperty or SecuredListProperty interface conform to the specifications.

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 registers an AuthListener with the AllJoyn framework that provides the default passcode ("000000") when authentication is requested.
4. The test device uses the Introspect() method on the Introspectable interface to locate all bus objects implementing the org.alljoyn.ControlPanel.ListProperty or org.alljoyn.ControlPanel.SecuredListProperty interface.
5. For each matching bus object found, the test device performs the following steps:
   a. Retrieve the Version property.
   b. Retrieve the States property.

   c. Retrieve the OptParams property.
   d. Retrieve the Value property.
6. The test device leaves the session.

Expected results

- If no bus objects implementing the ListProperty or SecuredListProperty interface are found, a note is added to this effect.
- For each bus object implementing the ListProperty or SecuredListProperty interface:
  - ❐ The Version property is 1
  - ❐ The States property is 0x00 or 0x01
  - ❐ The OptParams property is set as follows:
    - If OptParams contains a key with '0', the value type is a string.
    - If OptParams contains a key with '1', the value type is a uint32.
    - If OptParams contains a key with '2' (the hints field), the value type is an array of uint16, the array has a size of one, and the value of the one element must be 1 (DynamicSpinner).
  - ❐ The Value property containing a list of records where the recordID is unique within the list.
  - ❐ The bus object is a descendant of a bus object implementing one of the following interfaces:
    - org.alljoyn.ControlPanel.Container
    - org.alljoyn.ControlPanel.SecuredContainer

3.8 ControlPanel-v1-08: Verify all NotificationAction bus objects

Objective

Verify that all bus objects implementing the NotificationAction interface conform to the specifications.

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 registers an AuthListener with the AllJoyn framework from the AllJoyn framework that provides the default passcode ("000000") when authentication is requested.

4. The test device uses the Introspect() method on the Introspectable interface to locate all bus objects implementing the org.alljoyn.ControlPanel.NotificationAction interface.
5. For each matching bus object found, the test device performs the following steps:
   a. Retrieve the Version property.
   b. Use the Introspect() method on the Introspectable interface to locate child bus objects which implement the org.alljoyn.ControlPanel.Container or org.alljoyn.ControlPanel.SecuredContainer interface.
6. The test device leaves the session.

Expected results

- If no bus objects implementing the NotificationAction interface are found, a note is added to this effect.
- For ach bus object implementing the NotificationAction interface:
  - The object's path matches the pattern "/NotificationPanel/[^/]+/[^/]+ "
  - The Version property is 1
  - At least one child bus object implements the org.alljoyn.ControlPanel.Container or org.alljoyn.ControlPanel.SecuredContainer interface.

3.9 ControlPanel-v1-09: Verify all HTTPControl bus objects

Objective

Verify that all bus objects implementing the HTTPControl interface conform to the specifications.

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 locate all bus objects implementing the org.alljoyn.ControlPanel.HTTPControl interface.
4. For each matching bus object found, the test device performs the following steps:
   a. Retrieve the Version property.
   b. Call the GetRootURL() method.
   c. Make an HTTP request for the root control page URL returned from the GetRootURL method.
5. The test device leaves the session.

Expected results

- If no bus objects implementing the HTTPControl interface are found, a note is added to this effect.
- For each bus object implementing the HTTPControl interface:
  - The object's path matches the pattern "/Control/[^/]+/HTTPControl"
  - The Version property is 1
  - The GetRootURL method returns a valid URL.
  - An HTTP request for the root control page URL returns a valid HTTP status code.

3.10 ControlPanel-v1-10: Verify all secured ControlPanel bus objects

Objective

Verify that all bus objects implementing one of the Secured ControlPanel interfaces require security as per the specficiation.

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 locate all bus objects implementing one of the following Secured ControlPanel interfaces:
   - SecuredContainer
   - SecuredProperty
   - SecuredAction
   - SecuredDialog
   - SecuredListProperty
4. For each matching bus object found, the test device performs the following steps:
   a. Clear all stored keys from the key store.
   b. Leave any existing session.
   c. Register an AuthListener with the AllJoyn framework that provides the passcode "123456" when authentication is requested.
   d. Join a new session with the application at the port specified in the received About announcement.
   e. Attempt to retrieve the Version property from the bus object
5. The test device leaves the session.

Expected results

- If no bus objects implementing any of the Secured ControlPanel interfaces are found, a note is added to this effect.
- Each attempt to retrieve the Version property from a Secured ControlPanel interface fails and the completed method is called on the registered AuthListener indicating that the authentication failed.

4 Interface Data

This chapter lists the interface data supported by the Control Panel service framework.

NOTE

All Control Panel service framework interfaces are mandatory. Metadata information is provided in the *AllJoyn™ Control Panel Service Framework Interface Specification*.

The following sections note which interfaces use metatdata and supported data types.

4.1 ControlPanel interface

While this interface has no methods or signals, it is considered mandatory.

4.2 Container interface

The OptParams property makes use of Container widget metadata detailed in this section.

4.2.1 MetadataChanged signal

This signal is considered mandatory.

4.3 SecuredContainer interface

See Container interface.

4.4 Property interface

The OptParams property makes use of Property widget metadata.

The Values property makes use of the supported data types.

4.4.1 ValueChanged signal

Parameter	Mandatory
newValue	yes

4.4.2 MetadataChanged signal

This signal is considered mandatory.

4.5 SecuredProperty interface

See Property interface.

4.6 LabelProperty interface

The OptParams property makes use of LabelProperty widget metadata.

4.6.1 MetadataChanged signal

This signal is considered mandatory.

4.7 Action interface

The OptParams property makes use of Action widget metadata.

4.7.1 Exec method

This method is considered mandatory.

4.7.2 MetadataChanged signal

This signal is considered mandatory.

4.8 SecuredAction interface

See Action interface.

4.9 NotificationAction interface

4.9.1 Dismiss signal

This signal is considered mandatory.

4.10 Dialog interface

The OptParams property makes use of Dialog widget metadata.

4.10.1 Action1 method

This method is considered mandatory.

4.10.2 Action2 method

This method is considered mandatory.

4.10.3 Action3 method

This method is considered mandatory.

4.10.4 MetadataChanged signal

This signal is considered mandatory.

4.11 SecuredDialog interface

See Dialog interface.

4.12 ListProperty interface

The OptParams property makes use of ListProperty metadata.

4.12.1 Add method

This method is considered mandatory.

Prepare the input form to add a new record to the list. UI requirements follow:

- The controller must present an OK button and tie it to the Confirm() method call. Completing the add action on the input form will add the new record to the list.
- The controller must present a Cancel button and tie to the Cancel() method call to allow for discarding the operation.

4.12.2 Delete method

Parameter	**Mandatory**
recordID	yes

Prepare the form for view the record prior to the delete action. UI requirements follow:

- The controller must present an OK button and tie it to the Confirm() method call. A confirm action deletes the record from the list.
- The controller must present a Cancel button and tie to the Cancel() method call to allow for discarding the operation.

4.12.3 View method

Parameter	Mandatory
recordID	yes

Prepare the display form to view the record identified by the recordID.

The controller must present an OK button to dismiss the view form.

4.12.4 Update method

Parameter	Mandatory
recordID	yes

Prepare the input form to view the record identified by the recordID and allow the end user to modify the fields. UI requirements follow:

- The controller must present an OK button and tie it to the Confirm() method call. A confirm action deletes the record from the list.
- The controller must present a Cancel button and tie to the Cancel() method call to allow for discarding the operation.

4.12.5 Confirm method

This method is considered mandatory.

Confirm the action and save the change requested.

The controller must present an OK button to dismiss the view form.

4.12.6 Cancel method

This method is considered mandatory.

The controller must present a Cancel button to dismiss the input form.

4.12.7 ValueChanged signal

This signal is considered mandatory.

4.12.8 MetadataChanged signal

This signal is considered mandatory.

4.13 SecuredListProperty interface

See ListProperty interface.

4.14 HTTPControl interface

4.14.1 GetRootURL method

Parameter	Mandatory
url	yes