ALLSEEN ALLIANCE

AllJoyn™ Control Panel Service Framework Interface Definition

Version 14.06 Update 1

September 29, 2014

Contents

Figures

Tables

1 Introduction

1.1 Purpose

This document describes the specification of the Control Panel service framework's collection of interfaces. Implementing these interfaces provides the Control Panel service framework a mechanism to allow for a controller application like the AllJoyn™ On application to render the UI based on the controllee application's widget metadata. The controller application does not know the semantics of the remote widgets. It relies on the end user to understand and perform the actions on the rendered user interface. A typical action is selecting a field or pressing a button.

1.2 Scope

This document is targeted to developers of AllJoyn™ applications who want to build controller and controllee applications without using the tools provided by the Control Panel service framework.

1.3 Release history

Release version	Date	What changed
14.02	2/28/2014	The following interfaces were added: ■ ControlPanel.ControlPanel interface version 1 ■ ControlPanel.Container interface version 1 ■ ControlPanel.SecuredContainer interface version 1 ■ ControlPanel.Property interface version 1 ■ ControlPanel.SecuredProperty interface version 1 ■ ControlPanel.LabelProperty interface version 1 ■ ControlPanel.Action interface version 1 ■ ControlPanel.SecuredAction interface version 1 ■ ControlPanel.NotificationAction interface version 1 ■ ControlPanel.Dialog interface version 1 ■ ControlPanel.SecuredDialog interface version 1 ■ ControlPanel.ListProperty interface version 1 ■ ControlPanel.SecuredListProperty interface version 1 ■ ControlPanel.HTTPControl interface version 1
14.06	6/30/2014	No updates
14.06 Update 1	9/29/2014	■ Updated the document title and Overview chapter title (changed Specification to Definition). ■ Added the version release number to the title for version tracking.

Release version	Date	What changed
		■ Added a note in the Definition Overview chapter to address the AllSeen Alliance Compliance and Certification program. ■ Added a Mandatory column for method and signal parameters to support the AllSeen Alliance Compliance and Certification program.

1.4 References

Except for supporting information, the following are reference documents found on the AllSeen Alliance web site's Docs/Downloads section.

- *AllJoyn™ Framework Tutorial*
- *Introduction to AllJoyn™ Thin Library*
- *AllJoyn™ Data Type Signature*
- *RFC3339 (Date and Time on the Internet: Timestamps)*

1.5 Acronyms and terms

Term	Definition
AllJoyn device	A device that supports the AllJoyn framework and can connect to a personal network.
AllJoyn On application	The application running on the controller that displays and invokes the controls of another AllJoyn device.
Control panel	Collection of Widgets that allow a user to interact with a device. A control panel is defined and announced by a Controllee; and discovered and displayed by a Controller. A device can have more than one, and can be defined on a per-language basis.
Controllee	An AllJoyn application that advertises its Control Panel interfaces, so that other AllJoyn devices may control it.
Controller	An AllJoyn application that controls another AllJoyn device that advertises its Control Panel interfaces

2 Definition Overview

The Control Panel interfaces must be implemented by an application on a controllee. *Figure 1* illustrates the relationship between a controllee app and a controller app.

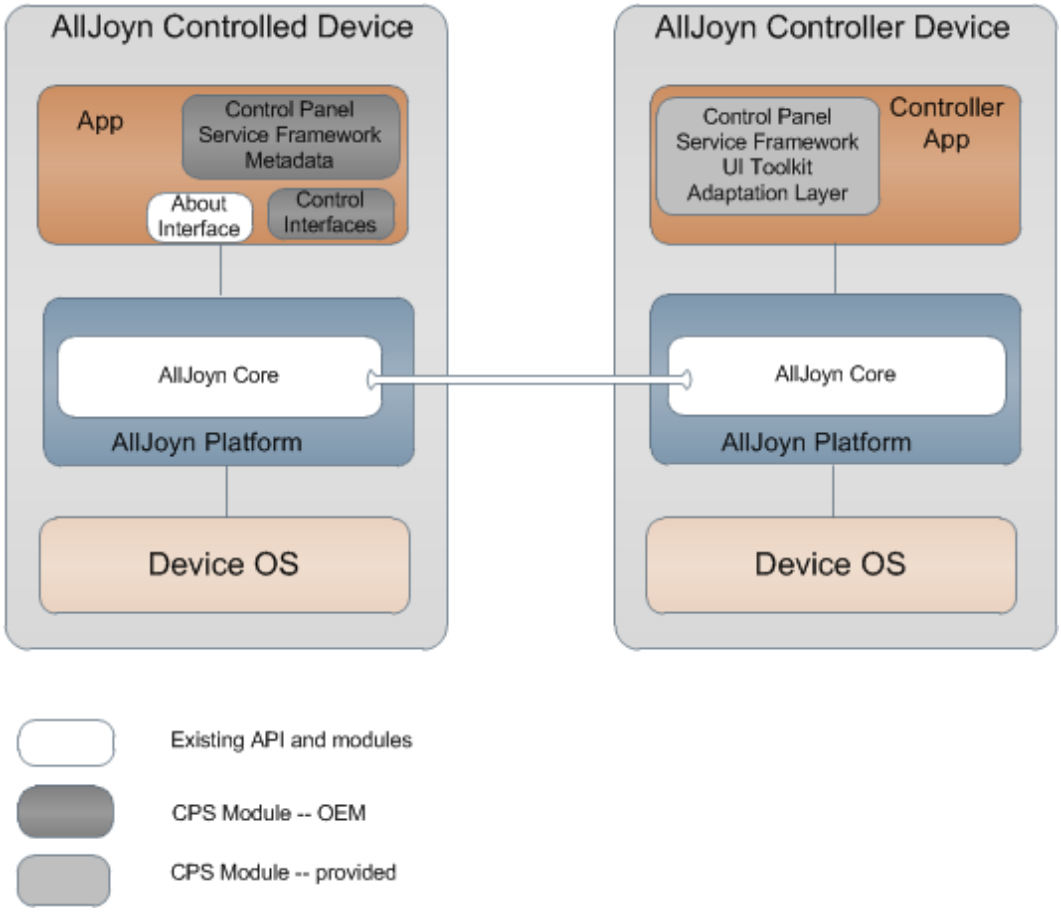

Figure 1: Control Panel service framework architecture within the AllJoyn framework

The OEM is responsible for writing the Control interfaces and the Control Panel service framework metadata.

The UI Toolkit Adaption Layer, a library used to map the metadata to platform-specific UI elements, is made available as part of the Control Panel service framework release.

Note All methods and signals are considered mandatory to support the AllSeen Alliance Compliance and Certification program. Individual parameters for a given method or signal may be considered mandatory or optional, and are specified accordingly in this document.

3 Discovery

Controllees are discovered via an AllJoyn announcement. Each AllJoyn device uses the About feature to announce basic app information like app name, device name, manufacturer, and model number. The announcement also contains the list of object paths and service interfaces to allow the controller to determine whether the controllee provides functionality of interest.

The About announcement is propagated using a sessionless signal.

4 Call Flows

4.1 Static Control Panel flow

Figure 2 illustrates a typical call flow for a control panel that does not change once rendered.

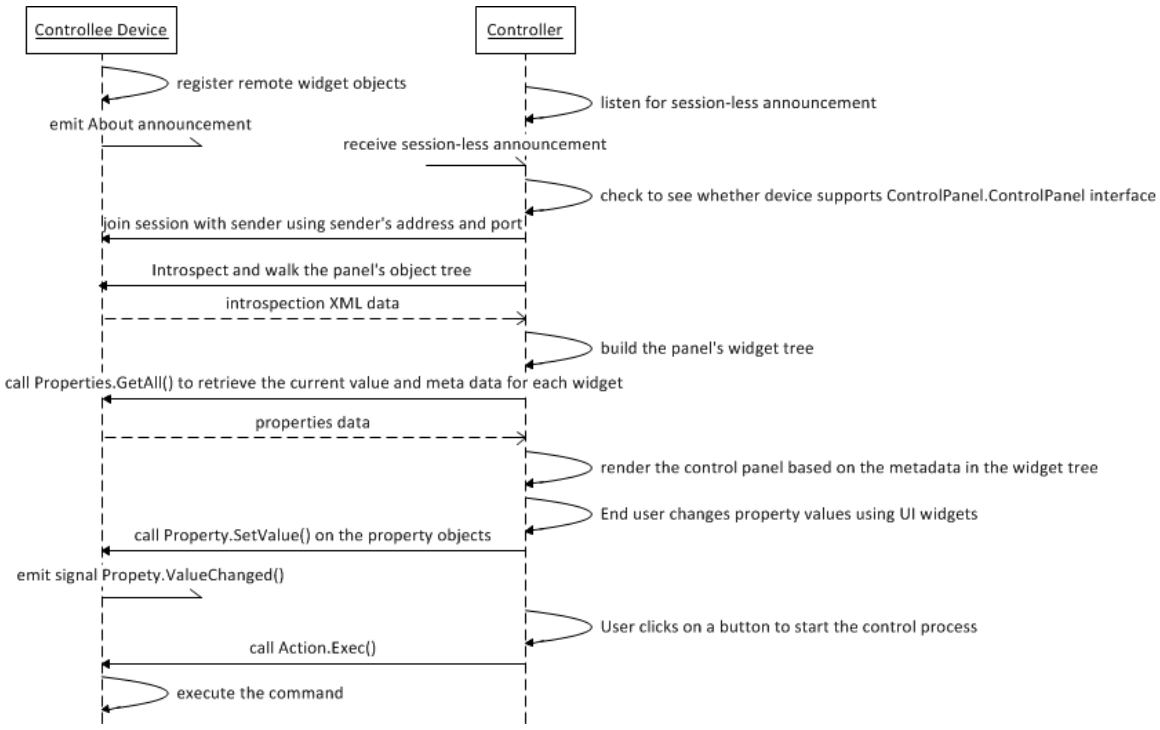

Figure 2: Static Control Panel flow

4.2 Dynamic Control Panel flow

Figure 3 illustrates a call flow for a control panel that changes as the end user interacts with the widgets.

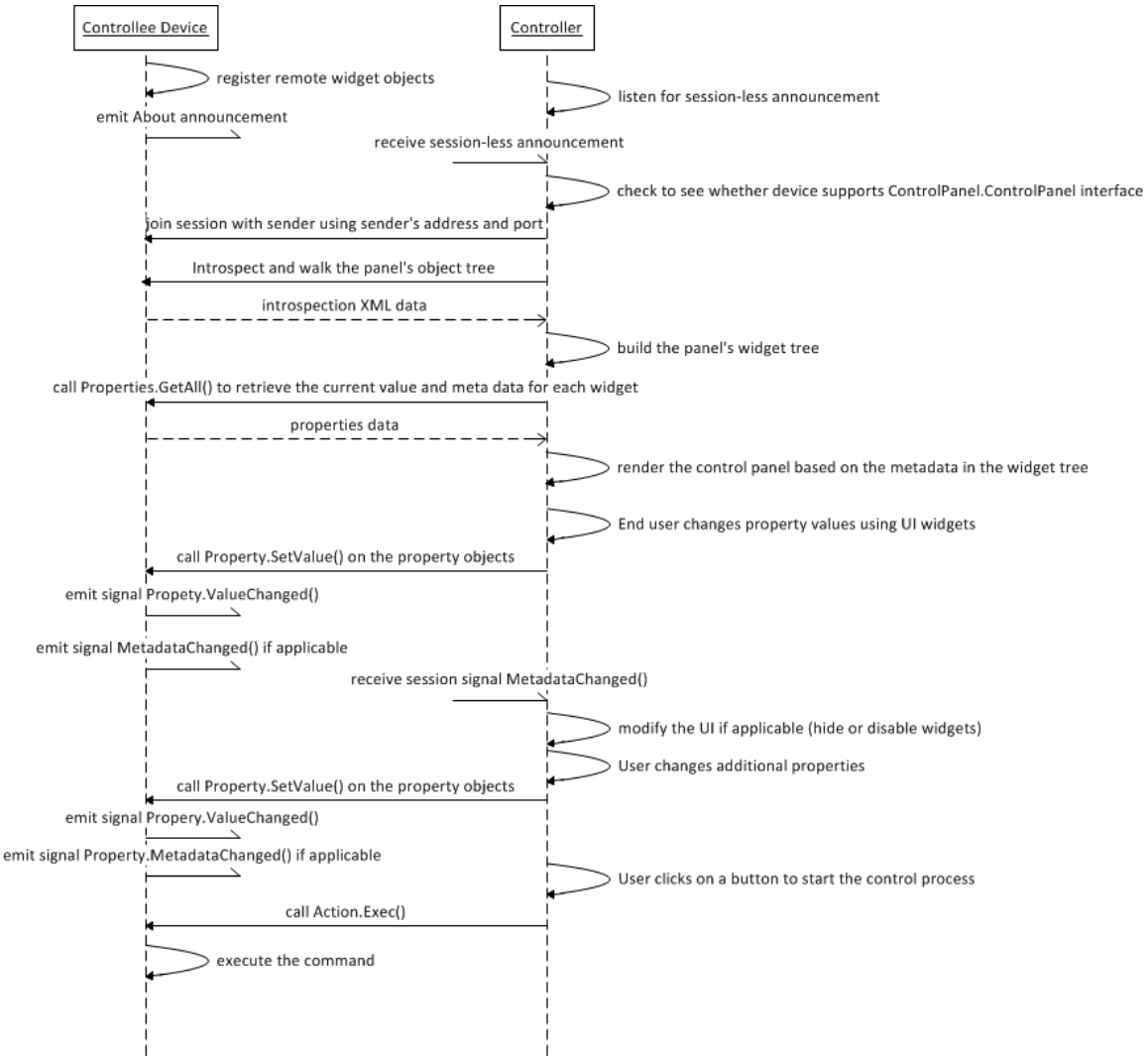

Figure 3: Dynamic Control Panel flow

5 Error Handling

The method calls in the Control Panel interfaces use the AllJoyn error message handling feature (ER_BUS_REPLY_IS_ERROR_MESSAGE) to set the error name and error message.

Table 1 lists the possible errors raised by the Control Panel service framework interfaces.

Table 1: Control Panel service framework interface errors

Error name	Error message
org.alljoyn.Error.OutOfRange	Value out of range
org.alljoyn.Error.InvalidState	Invalid state
org.alljoyn.Error.InvalidProperty	Invalid property
org.alljoyn.Error.InvalidValue	Invalid value
org.alljoyn.Error.MethodNotAllowed	Method call not allowed

6 BusObject Map

6.1 BusObject structure

Figure 4 shows the tree structure diagram that represents the basic organization of AllJoyn objects used in the support of the Control Panel service framework. A control panel is implemented using several AllJoyn objects.

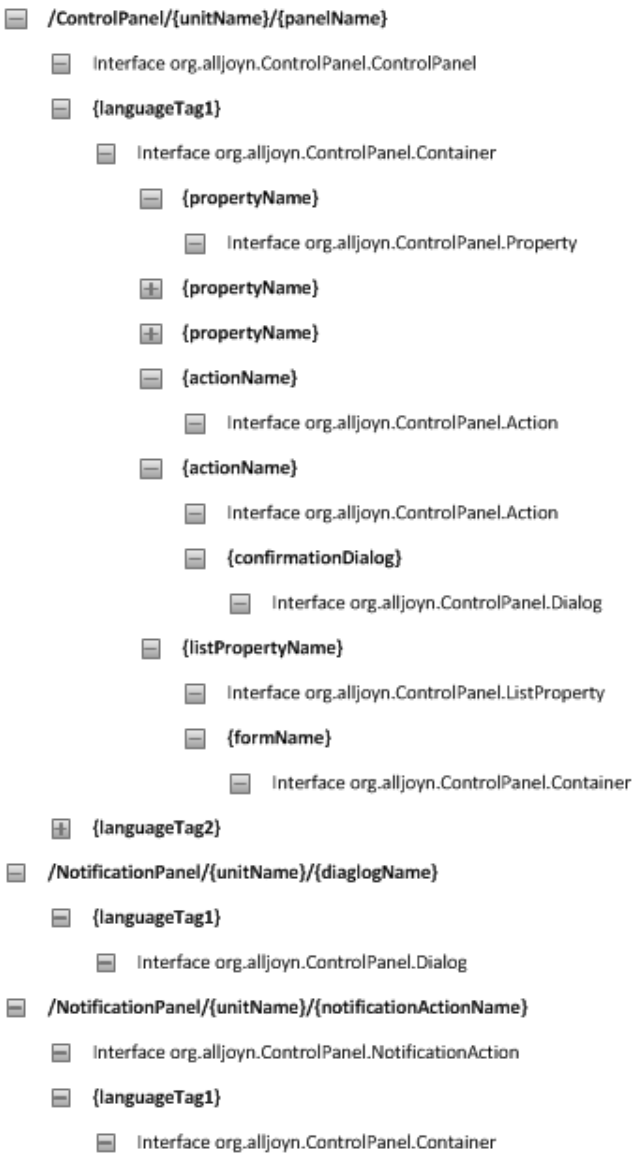

Figure 4: BusObject map

The objects are organized to support multiple units and multiple languages. Only the top-level panels should be listed in the announcement.

Since the IETF language tag allows for hyphen (-) which is not allowed in the bus object path, any language tag in the object path replaces a hyphen (-) with an underscore (_).

In addition to the control panels, the Control Panel service framework can also support other panels such as a notification panel. These panels are not required to be advertised in the announcement.

It is the controller's job to introspect and walk the object tree of a control panel to retrieve all the metadata for that control panel.

6.2 BusObject examples

The following subsections illustrate a few sample BusObject map structures.

6.2.1 Washing machine example

Figure 5 illustrates a sample bus object map for a washing machine.

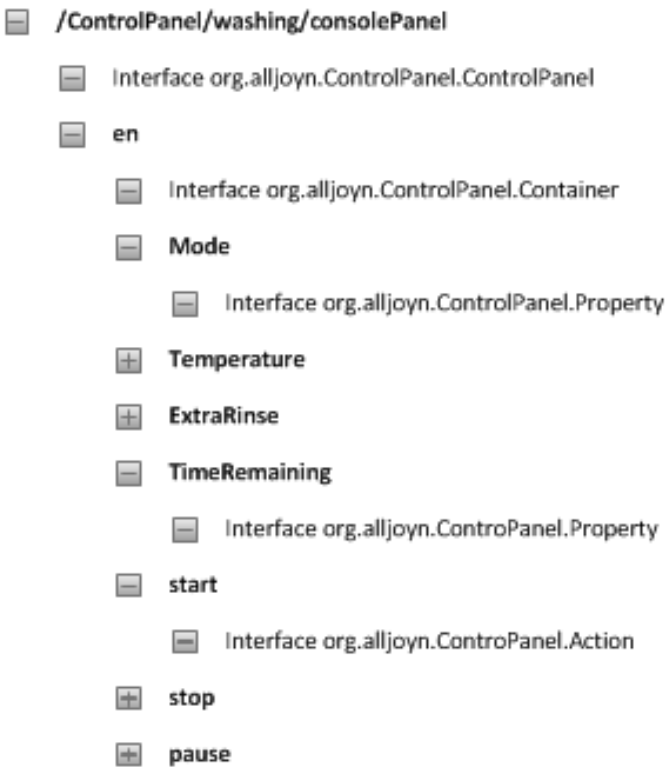

Figure 5: Washing machine example

6.2.2 Sprinkler system example

Figure 6 illustrates a sample bus object map for a sprinkler system.

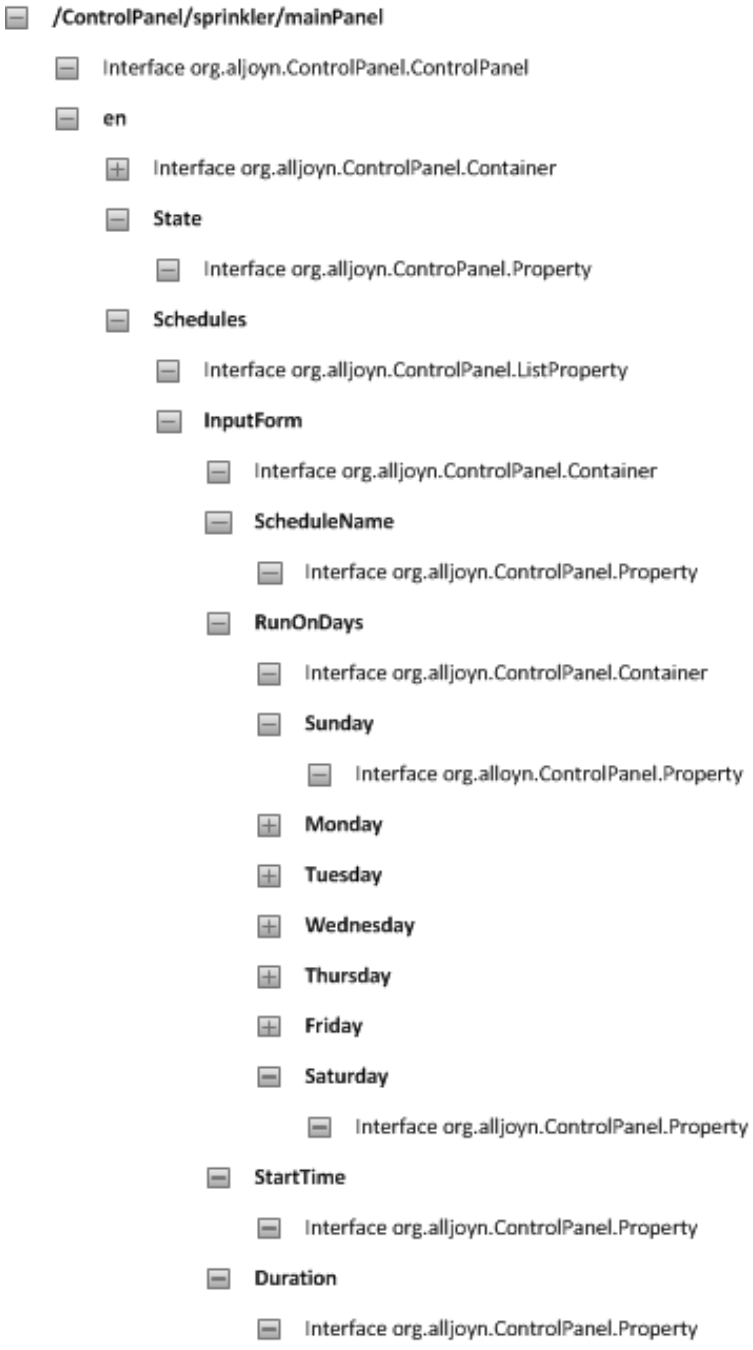

Figure 6: Sprinkler system example

7 Control Panel Service Framework Interfaces

The Control Panel service framework comprises a set of interfaces as illustrated in *Figure 4*.

8 ControlPanel Interface

This interface indicates whether the object is a control panel. This object will support at least one language. The service only needs to advertise this type of object in the About announcement. No other objects in the Control Panel service framework tree should be announced.

Note It's the responsibility of the controller to introspect the children objects to locate the corresponding root container of the given panel for the specific language code.

8.1 Interface name

Interface name	Version	Secured	Object Path
org.alljoyn.ControlPanel.ControlPanel	1	no	/ControlPanel/{unit)/{panelName} Example: /ControlPanel/washing/consolePanel

8.2 Properties

Property name	Signature	List of values	Writable	Description
Version	q	Positive integers	no	Interface version number

8.3 Introspect XML

The following XML defines the ControlPanel interface.

```
<node
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="https://www.allseenalliance.org/schemas/introspect.xsd">
<interface name="org.alljoyn.ControlPanel.ControlPanel">
  <property name="Version" type="q" access="read"/>
</interface>
</node>
```

9 Container Interface

This interface provides all the metadata to guide the controller to render the UI for a container widget.

9.1 Interface name

Interface name	Version	Secured	Object path
org.alljoyn.ControlPanel.Container	1	no	/ControlPanel/{unit)/{panelName}/{language}/…/{containerName} Examples: /ControlPanel/washing/consolePanel/en /ControlPanel/sprinkler/mainPanel/en/Schedules/InputForm/RunOnDays
org.alljoyn.ControlPanel.SecuredContainer	1	yes	/ControlPanel/{unit)/{panelName}/{language}/…/{containerName}

9.2 Properties

Property name	Signature	List of values	Writable	Description		
Version	q	Positive integers	no	Interface version number		
States	u	Bit mask	no	Bit mask for various widget states		
				Mask	**Name**	**Description**
				0x01	enabled	Indicates whether the widget is enabled. A disabled widget should be grayed out or invisible.
OptParams	a{qv}		no	The metadata dictionary. See *Container widget metadata* for more information.		

9.2.1 Container widget metadata

The following table defines the metadata of a container widget.

Dictionary key	Field name	Value type	Description
0	Label	s	Label

Dictionary key	Field name	Value type	Description
1	bgColor	u	Background color expressed as RGB value. If not specified, then the background color of the enclosing container is used.
2	layoutHints	aq	Layout hints. See *Container widget layout hints* for more information.

9.2.2 Container widget layout hints

The following table lists layout hints for a container widget.

Hint ID	Hint name	Description
1	Vertical Linear	A layout that aligns all components in a vertical direction
2	Horizontal Linear	A layout that aligns all components in a horizontal direction

9.3 Methods

No methods are exposed by this interface.

9.4 Signals

Signal name	Mandatory	Sessionless	Description
MetadataChanged	yes	no	The metadata has changed. This can occur due to changes in any of the property objects.

9.5 Introspect XML

The following XML defines the Container interface.

```
<node
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="https://www.allseenalliance.org/schemas/introspect.xsd">
<interface name="org.alljoyn.ControlPanel.Container">
  <property name="Version" type="q" access="read"/>
  <property name="States" type="u" access="read"/>
  <property name="OptParams" type="a{qv}" access="read"/>
  <signal name="MetadataChanged" />
</interface>
</node>
```

10 Property Interface

This interface provides the control mechanism for the property widget. Each widget is represented by an AllJoyn object implementing this interface.

10.1 Interface name

Interface name	Version	Secured	Object path
org.alljoyn.ControlPanel.Property	1	no	/ControlPanel/{unit}/{panel}/{language} /…/ {object name} Examples: /ControlPanel/washing/consolePanel/en /Mode /ControlPanel/sprinkler/mainPanel/en/ Schedules/InputForm/ScheduleName
org.alljoyn.ControlPanel.SecuredProperty	1	yes	/ControlPanel/{unit}/{panel}/…/{object name}

10.2 Properties

Property name	Signature	List of Values	Writable	Description		
Version	q	Positive integers	no	Interface version number		
States	u	Bit mask	no	Bit mask for various widget states		
				Mask	**Name**	**Description**
				0x01	enabled	Indicates whether the widget is enabled. A disabled widget should be grayed out or invisible.
				0x02	writable	Indicates whether the property is writable
OptParams	a{qv}		no	Metadata dictionary. See *Property widget metadata* for more information.		
Value	v		yes	Actual value of the property. When modifying the property, the device may give the AllJoyn error *org.alljoyn.Error.MethodNotAllowed* if the property is ReadOnly.		

Property name	Signature	List of Values	Writable	Description
				The supported data types are listed in *Supported data types*

10.2.1 Property widget metadata

The following table lists property widget metadata.

Dictionary key	Field name	Value type	Description
0	label	s	The label.
1	bgColor	u	Background color expressed as RGB value. If not specified, then the background color of the enclosing container is used.
2	hints	aq	The widget rendering hints. See *Property widget hints* for more information.
3	unitOfMeasure	s	Unit of measurement
4	constrainToValue	a(vs)	Constraint on the value as a list of values. Any value of the property must match one of the values in this list. See *List of values* for more information.
5	range	vvv	Constraint on the value as a range; a value of this property must stay within the range. See *Property widget ranges* for more information.

10.2.2 Property widget hints

Hint ID	Hint name	Description
1	Switch	Two-state buttons allowing the end-user to toggle the state of a single settings option.
2	CheckBox	Widget for multi-select. It allows the end user to select multiple options from a list.
3	Spinner	Widget for single-select. It allows the end user to select a single option from a list.
4	RadioButton	Widget for single-select. It allows the end user to select a single option from a list.
5	Slider	Allows the end user to select a value from a continuous or discrete range. The appearance is linear, either horizontal or vertical.
6	TimePicker	Allows the end user to specify a time value
7	DatePicker	Allows the end user to specify a date value
8	NumberPicker	Allows the end user to specify a numeric value

Hint ID	Hint name	Description
9	NumericKeypad	Provides the end user with a numeric entry field and buttons for 0-9 digits, to enter a numeric value. The developer must know the min/max number of digits allowed in the entry field.
10	RotaryKnob	An alternate way to represent a slider
11	TextLabel	Read-only text label
12	NumericView	Provides a read-only, numeric field with an optional label and numbers. For example, a washing machine display shows the time remaining for wash is 35:00 minutes.
13	EditText	Provides the end user with a text entry field and keyboard. The developer must know the min/max number of letters allowed in the entry field.

10.2.3 Supported data types

The following table lists the data types supported by the Control Panel service framework.

<table>
<tr><th>Category</th><th colspan="4">Supported data types</th></tr>
<tr><td>Scalar types</td><td colspan="4">■ BOOLEAN – b
■ BYTE – y
■ BYTE ARRAY – ay
■ Numeric types:
❒ INT16 – n
❒ UINT16 – q
❒ INT32 – i
❒ UINT32 – u
❒ INT64 – x
❒ UINT64 – t
❒ DOUBLE – d
■ STRING – s</td></tr>
<tr><td>Composite types</td><td colspan="4">All composite data type must have the following signature—q(type) —where the first value is an enum value indicating the composite type</td></tr>
<tr><td></td><td>Composite type enum</td><td>Composite type name</td><td>Signature</td><td>Description</td></tr>
<tr><td></td><td>0</td><td>Date</td><td>q(qqq)</td><td>Date type per RFC3339. There are three fields:
■ date-mday (1-31)
■ date-month (1-12)
■ date-fullyear (4-digit year)</td></tr>
</table>

Category	Supported data types			
	1	Time	q(qqq)	Time type following RFC3339. There are three fields: ■ time-hour (0-23) ■ time-minute (0-59) ■ time-second (0-59)
Collection of records	Array of records of only scalar and supported composite types. All records in the array must be of the same record type.			

10.2.4 List of values

A list of values is an array of structs.

Field name	Data type	Description
Value	v	Value with the same data type of the property
Label	s	Display label

10.2.5 Property widget ranges

The following table lists Property widget ranges.

Field name	Data type	Description
min	v	Minimum value with the same data type as that of the property
max	v	Maximum value with the same data type as that of the property
increment	v	Value to increment/decrement by. It has the same data type as that of the property

10.3 Methods

No methods are exposed by this interface.

10.4 Signals

Signal name	Mandatory	Sessionless	Description
MetadataChanged	yes	no	The metadata has changed. This can occur due to changes in any of the property objects.
ValueChanged	yes	no	The property's value has changed

10.5 Introspect XML

The following XML defines the Property interface.

```
<node
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="https://www.allseenalliance.org/schemas/introspect.xsd">
<interface name="org.alljoyn.ControlPanel.Property">
  <property name="Version" type="q" access="read"/>
  <property name="States" type="u" access="read"/>
  <property name="OptParams" type="a{qv}" access="read"/>
  <property name="Value" type="v" access="readwrite"/>
  <signal name="MetadataChanged" />
  <signal name="ValueChanged">
     <arg type="v"/>
  </signal>
</interface>
</node>
```

11 LabelProperty Interface

This interface provides the control mechanism for the label property widget (a text label). Each widget is represented by an AllJoyn object implementing this interface.

11.1 Interface name

Interface name	Version	Secured	Object path
org.alljoyn.ControlPanel.LabelProperty	1	no	/ControlPanel/{unit}/{panel}/{language}/…/ {object name} Example: /ControlPanel/airconditioner/console/Warning

11.2 Properties

Property name	Signature	List of values	Writable	Description		
Version	q	Positive integers	no	Interface version number		
States	u	Bit mask	no	Bit mask for various widget states.		
				Mask	**Name**	**Description**
				0x01	enabled	Indicates whether the widget is enabled. A disabled widget should be grayed out or invisible.
Label	s		no	Text label		
OptParams	a{qv}		no	Metadata dictionary. See *LabelProperty widget metadata* for more information.		

11.2.1 LabelProperty widget metadata

The following table lists LabelProperty widget metadata.

Dictionary key	Field name	Value type	Description
1	bgColor	u	Background color expressed as RGB value. If not specified, then the background color of the enclosing container is used.
2	hints	aq	The widget rendering hints. See *LabelProperty widget hints* for more information.

11.2.2 LabelProperty widget hints

The following table lists LabelProperty widget hints.

Hint ID	Hint name	Description
1	TextLabel	Read-only text label

11.3 Methods

No methods are exposed by this interface.

11.4 Signals

Signal name	Mandatory	Sessionless	Description
MetadataChanged	yes	no	The metadata has changed. This can occur due to changes in any of the property objects.

11.5 Introspect XML

The following XML defines the LabelProperty interface.

```
<node
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="https://www.allseenalliance.org/schemas/introspect.xsd">
<interface name="org.alljoyn.ControlPanel.LabelProperty">
  <property name="Version" type="q" access="read"/>
  <property name="States" type="u" access="read"/>
  <property name="Label" type="s" access="read"/>
  <property name="OptParams" type="a{qv}" access="read"/>
  <signal name="MetadataChanged" />
</interface>
</node>
```

12 Action Interface

This interface provides the control mechanism for the Action widget. Each Action widget is represented by an AllJoyn object implementing this interface. An action widget can optionally provide a confirmation dialog widget in its object sub-tree to allow for a pop-up dialog to appear whenever the UI presentation of this action is activated. The action taken on the confirmation dialog will take place instead of the Exec() method call for this Action widget.

12.1 Interface name

Interface name	Version	Secured	Object path
org.alljoyn.ControlPanel.Action	1	no	/ControlPanel/{unit}//{panel}/{language}/…/{object name} Example: /ControlPanel/washing/mainPanel/en/start
org.alljoyn.ControlPanel.SecuredAction	1	yes	/ControlPanel/{unit}/{panel}/{language}/…/{object name}

12.2 Properties

Property name	Signature	List of values	Writable	Description		
Version	q	Positive integers	no	Interface version number		
States	u	Bit mask	no	Bit mask for various widget states.		
				Mask	**Name**	**Description**
				0x01	enabled	Indicates whether the widget is enabled. A disabled widget should be grayed out or invisible.
OptParams	a{qv}		no	Metadata dictionary. See *Action widget metadata* for more information.		

12.2.1 Action widget metadata

The following table lists action widget metadata.

Dictionary key	Field name	Value type	Description
0	label	s	Label
1	bgColor	u	Background color expressed as RGB value. If not specified, then the background color of the enclosing container is used.
2	hints	aq	The widget rendering hints. See *Action widget hints* for more information.

12.2.2 Action widget hints

The following table lists Action widget hints.

Hint ID	Hint name	Description
1	ActionButton	Button associated with an action or a method call, for example “submit.”

12.3 Methods

12.3.1 Exec

Inputs

None.

Output

None.

Description

Execute the action command.

12.4 Signals

Signal name	Mandatory	Sessionless	Description
MetadataChanged	yes	no	The metadata has changed. This can occur due to changes in any of the property objects.

12.5 Introspect XML

The following XML defines the Action interface.

```
<node
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="https://www.allseenalliance.org/schemas/introspect.xsd">
<interface name="org.alljoyn.ControlPanel.Action">
```

```
  <property name="Version" type="q" access="read"/>
  <property name="States" type="u" access="read"/>
  <property name="OptParams" type="a{qv}" access="read"/>
  <signal name="MetadataChanged" />
  <method name="Exec"/>
</interface>
</node>
```

13 NotificationAction Interface

This interface indicates whether the object is a notification action object. A notification object is typically referenced in a notification message. Upon receiving such notification, the controller can generate the notification action panel based on the metadata provided by this type of object. This object is different from a regular control panel since it allows the controllee to send a signal to tell the controller to dismiss the panel.

This object supports at least one language. It's the responsibility of the controller to introspect the children objects to locate the corresponding root container of the given panel for the specific language code.

13.1 Interface name

Interface name	Version	Secured	Object path
org.alljoyn.ControlPanel.NotificationAction	1	no	/NotificationPanel/{unit}/{actionPanelName} Example: /NotificationPanel/washing/CycleCompleted

13.2 Properties

Property name	Signature	List of Values	Writable	Description
Version	q	Positive integers	no	Interface version number

13.3 Methods

No methods are exposed by this interface.

13.4 Signals

Signal name	Mandatory	Sessionless	Description
Dismiss	yes	no	The controller must dismiss this notification panel.

13.5 Introspect XML

The following XML defines the NotificationAction interface.

```
<node
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="https://www.allseenalliance.org/schemas/introspect.xsd">
<interface name="org.alljoyn.ControlPanel.NotificationAction">
  <property name="Version" type="q" access="read"/>
   <signal name="Dismiss" />
</interface>
</node>
```

14 Dialog Interface

This interface provides all the metadata to guide the controller to render the UI for a dialog widget. A dialog widget typically has a message and up to three action buttons.

14.1 Interface name

Interface name	Version	Secured	Object path
org.alljoyn.ControlPanel.Dialog	1	no	/ControlPanel/{unit}/{panelName}/{language}/…/{dialogName} Example: /ControlPanel/washing/mainPanel/en/Confirmation
org.alljoyn.ControlPanel.SecuredDialog	1	yes	/ControlPanel/{unit}/{panelName}/{language}/…/{dialogName}

14.2 Properties

Property name	Signature	List of values	Writable	Description		
Version	q	Positive integers	no	Interface version number		
States	u	Bit mask	no	Bit mask for various widget states.		
				Mask	**Name**	**Description**
				0x01	enabled	Indicates whether the widget is enabled. A disabled widget should be grayed out or invisible.
OptParams	a{qv}		no	Metadata dictionary. See *Dialog widget metadata* for more information.		
Message	q		no	Display message		
NumActions	q	1-3	no	Number of available actions		

14.2.1 Dialog widget metadata

The following table lists dialog widget metadata.

Dictionary key	Field name	Value type	Description
0	label	s	Label or title of the dialog
1	bgColor	u	Background color expressed as RGB value. If not specified then the background color of the enclosing container is used.
2	hints	aq	Layout hints. See *Dialog layout hints* for more information.
6	labelAction1	s	Label of the action1 widget
7	labelAction2	s	Label of the action2 widget
8	labelAction3	s	Label of the action3 widget

14.2.2 Dialog layout hints

The following table lists Dialog layout hints.

Hint ID	Hint name	Description
1	AlertDialog	Widget that combines a label, text data, and buttons in a single dialog box. A minimum of 1 button is required. A maximum of 3 buttons is supported.

14.3 Methods

14.3.1 Action1

Inputs

None.

Output

None.

Description

Execute the action number 1.

14.3.2 Action2

Inputs

None.

Output

None.

Description

Execute the action number 2. If the NumActions property is less than 2, the org.alljoyn.Error.MethodNotAllowed error will be raised.

14.3.3 Action3

Inputs

None.

Output

None.

Description

Execute the action number 3. If the NumActions property is less than 3, the org.alljoyn.Error.MethodNotAllowed error will be raised.

14.4 Signals

Signal name	Mandatory	Sessionless	Description
MetadataChanged	yes	no	The metadata has changed. This can occur due to changes in any of the property objects.

14.5 Introspect XML

The following XML defines the Dialog interface.

```
<node
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="https://www.allseenalliance.org/schemas/introspect.xsd">
<interface name="org.alljoyn.ControlPanel.Dialog">
  <property name="Version" type="q" access="read"/>
  <property name="States" type="u" access="read"/>
  <property name="OptParams" type="a{qv}" access="read"/>
  <property name="Message" type="s" access="read"/>
  <property name="NumActions" type="q" access="read"/>
  <signal name="MetadataChanged" />
  <method name="Action1"/>
  <method name="Action2"/>
  <method name="Action3"/>
</interface>
</node>
```

15 ListProperty Interface

This interface provides the control mechanism for the list property widget. A list property widget holds a list of records and a container representing the UI of the record display/input form.

15.1 Interface name

Interface name	Version	Secured	Object path
org.alljoyn.ControlPanel.ListProperty	1	no	/ControlPanel/{unit}/{language}/{panel}/{object name} Example: /ControlPanel/sprinkler/mainPanel/en/Schedules
org.alljoyn.ControlPanel.SecuredListProperty	1	yes	/ControlPanel/{unit}/{language}/{panel}/…/{object name}

15.2 Properties

Property name	Signature	List of values	Writable	Description		
Version	q	Positive integers	no	Interface version number		
States	u	Bit mask	n	Bit mask for various widget states.		
				Mask	**Name**	**Description**
				0x01	enabled	Indicates whether the widget is enabled. A disabled widget should be grayed out or invisible.
OptParams	a{qv}		no			Metadata dictionary. See *ListProperty widget metadata* for more information.
Value	a{qs}		no			List of records. Each record in the list holds the following fields: ■ recordID (q): the record ID ■ label (s): the label to display on the list

Property name	Signature	List of values	Writable	Description		
						The record data are not exposed in this property. The View() method call can be used to view each record.

15.2.1 ListProperty widget metadata

The following table lists ListProperty widget metadata.

Dictionary key	Field name	Value type	Description
0	label	s	Label
1	bgColor	u	Background color expressed as RGB value. If not specified, then the background color of the enclosing container is used
2	hints	aq	The widget rendering hints. See *List Property widget hints* for more information.

15.2.2 List Property widget hints

The following table lists ListProperty widget hints.

Hint ID	Hint name	Description
1	DynamicSpinner	Widget that allows the end user to select an option from a list, add a new option, delete an option, and update an option.

15.3 Methods

15.3.1 Add

Inputs

None.

Output

None.

Description

Prepare the input form for adding 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.

15.3.2 Delete

Inputs

Parameter name	Mandatory	Signature	Description
recordID	yes	q	The record ID

Output

None.

Description

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.

15.3.3 View

Inputs

Parameter	Mandatory	Signature	Description
recordID	yes	q	The record ID

Output

None.

Description

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

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

15.3.4 Update

Inputs

Parameter	Mandatory	Signature	Description
recordID	yes	q	The record ID

Output

None.

Description

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 updates the given record with new information.
- The controller must present a Cancel button and tie to the Cancel() method call to allow for discarding the operation.

15.3.5 Confirm

Inputs

None.

Output

None.

Description

Confirm the action and save the change requested.

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

15.3.6 Cancel

Inputs

None.

Output

None.

Description

Cancel the current action.

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

15.4 Signals

Signal name	Mandatory	Sessionless	Description
ValueChanged	yes	no	The property's value has changed. Because the list data can be large, the signal does not send the current value.

Signal name	Mandatory	Sessionless	Description
MetadataChanged	yes	no	The metadata has changed. This can occur due to changes in any of the property objects.

15.5 Introspect XML

The following XML defines the ListProperty interface.

```
<node
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="https://www.allseenalliance.org/schemas/introspect.xsd">
<interface name="org.alljoyn.ControlPanel.ListProperty">
  <property name="Version" type="q" access="read"/>
  <property name="States" type="u" access="read"/>
  <property name="OptParams" type="a{qv}" access="read"/>
  <property name="Value" type="a(qs)" access="read"/>
  <method name="Add"/>
  <method name="Delete">
     <arg name="recordID" type="q" direction="in"/>
  </method>
  <method name="View">
     <arg name="recordID" type="q" direction="in"/>
  </method>
  <method name="Update">
     <arg name="recordID" type="q" direction="in"/>
  </method>
  <method name="Confirm">
  </method>
  <method name="Cancel">
  </method>

  <signal name="MetadataChanged"/>
  <signal name="ValueChanged"/>
</interface>
</node>
```

16 Support of Existing HTTP Control Pages

Should a device already have HTTP control pages hosted on the device itself, it can advertise those pages using the following interface.

16.1 HTTPControl interface

This interface provides all the information about the hosted HTTP control pages on the device.

16.1.1 Interface name

Interface name	Version	Secured	Object path
org.alljoyn.ControlPanel.HTTPControl	1	no	/Control/{unit}/HTTPControl

16.1.2 Properties

Property name	Signature	List of values	Writable	Description
Version	q	Positive integers	no	Interface version number

16.1.3 Methods

16.1.3.1 GetRootURL

Inputs

None.

Output

Parameter name	Mandatory	Return signature	Description
url	Parameter	s	Root URL of the control pages

Description

Retrieve the URL of the root control page.

16.1.4 Signals

There is no signal in this interface.

16.1.5 Introspect XML

The following XML defines the HTTPControl interface.

```
<node
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="https://www.allseenalliance.org/schemas/introspect.xsd">
<interface name="org.alljoyn.Control.HTTPControl">
  <property name="Version" type="q" access="read"/>
  <method name="GetRootURL">
     <arg name="url" type="s" direction="out"/>
  </method>
 </interface>
</node>
```