ALLSEEN ALLIANCE

AllJoyn™ Configuration Service Framework Interface Definition

Version 14.06 Update 1

September 29, 2014

Contents

Figures

Tables

1 Introduction

1.1 Purpose

This document provides the specification for the AllJoyn™ Config interface. The Config interface is a secure interface that provides the functionality to perform device-specific configuration and actions. It is expected that an OEM's developed application for the device (referred to as System App) will bundle this service framework.

1.2 Scope

This document is targeted for OEMs who develop AllJoyn-enabled applications that configure and perform device specific actions. This interface specification is useful if the OEMs intend to develop or extend the Configuration service framework on their own.

1.3 Release history

Release version	Date	What changed
14.02	2/28/2014	Config interface version 1 was added.
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 release version number to the title for version tracking. ■ 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. ■ Added configData output parameter information to the GetConfigurations method.

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*
- *RFC 5646: Tags for Identifying Languages*

2 Definition Overview

The Configuration service framework exposes device-specific methods such as restart and factory reset, device passcode, and device-specific settable attributes such as friendly name and default language. It is expected that OEM of the device would take this service framework and bundle it with a single application (system app). The enforcement of singleton instance of the Configuration service framework must be performed using explicit guidelines provided to OEMs and application developers regarding the usage of the Configuration service framework.

Figure 1 illustrates the relationship between software stack on the device hosting the AllJoyn service framework and the device hosting the AllJoyn client application.

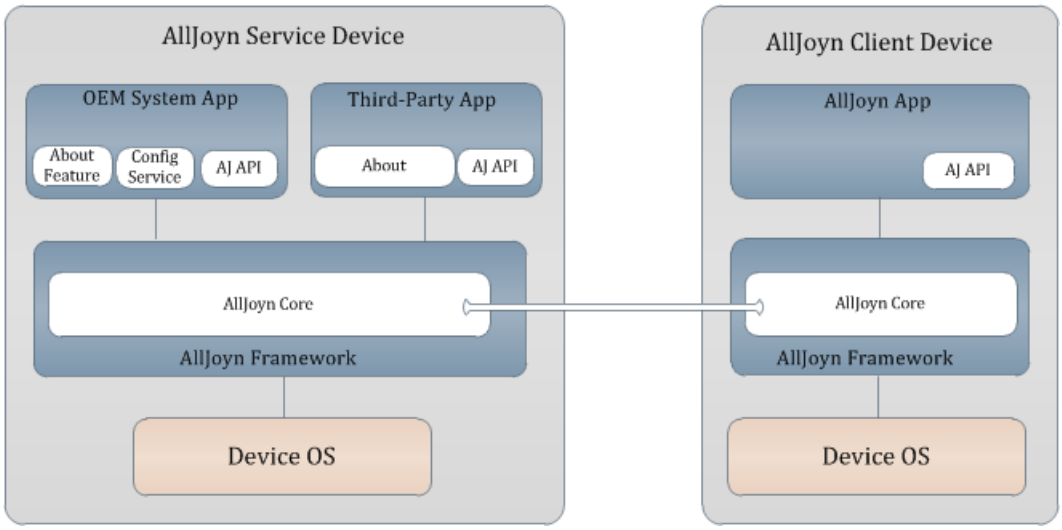

Figure 1: Configuration service framework architecture within the AllJoyn framework

Figure 1 describes the scope of Configuration service framework and About feature in a multiple applications-per-device scenario. The following system behavior should be noted:

- The system application bundles the Configuration service framework and provides a remote mechanism to invoke device-specific configuration.
- It could be that OEMs provide equivalent (as exposed by the Configuration service framework) functionality via the local user interface.

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 Typical Call Flows

This section highlights call flows that involve the Configuration service framework. The system app on the AllJoyn service framework device is involved in these call flows.

3.1 Device configuration change

Figure 2 illustrates a sample call flow where an Alljoyn app executing on an AllJoyn client device discovers the Configuration service framework via announcement and subsequently performs methods as specified in the Config interface to retrieve and update configuration data. See *Config Interface* for complete details.

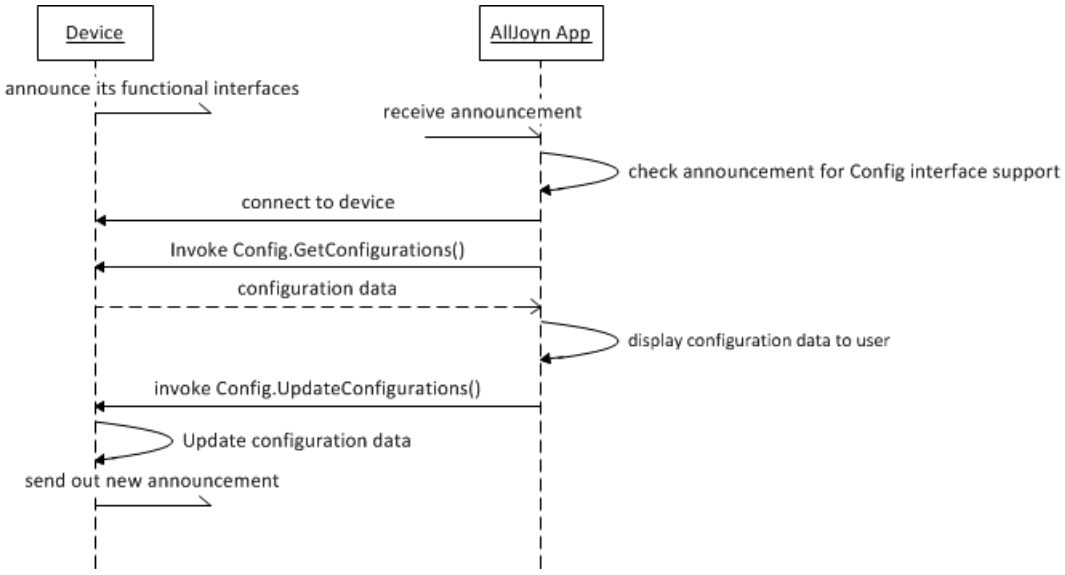

Figure 2: Device configuration change call flow

3.2 Device factory reset

Figure 3 illustrates a sample call flow where an Alljoyn app executing on an AllJoyn client device discovers the Configuration service framework via announcement, and subsequently performs methods as specified in the Config interface to retrieve the configuration data and perform factory reset action if needed. See *Config Interface* for complete details.

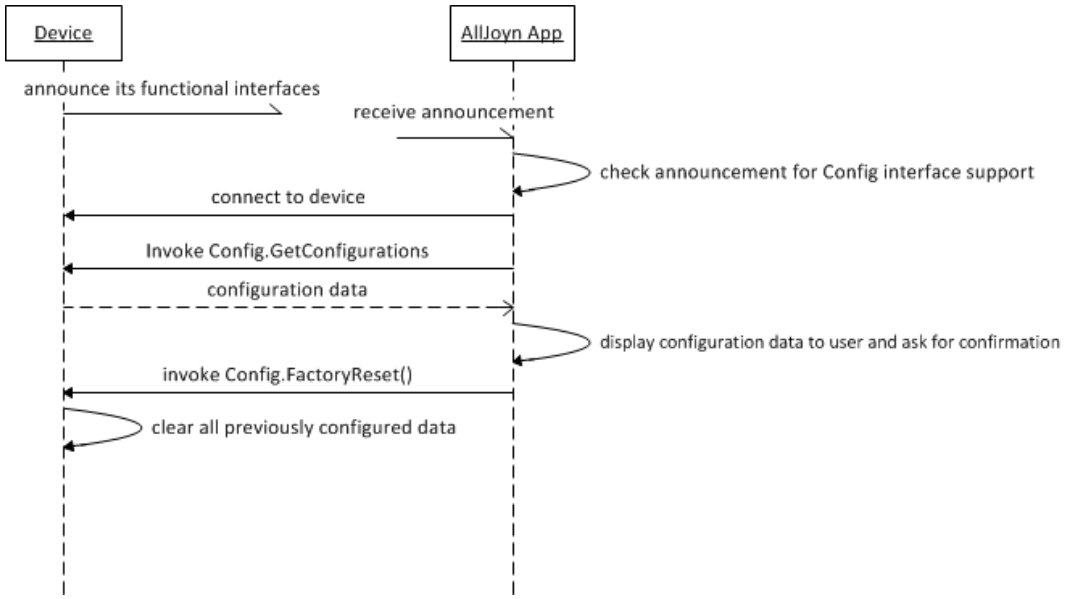

Figure 3: Device factory reset call flow

3.3 Error handling

The method calls in the Config interface 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 Config interface.

Table 1: Configuration service framework interface errors

Error name	Error message
org.alljoyn.Error.InvalidValue	Invalid value
org.alljoyn.Error.FeatureNotAvailable	Feature not available
org.alljoyn.Error.LanguageNotSupported	The language specified is not supported

4 Config Interface

4.1 Interface name

Interface Name	Version	Secured	Object path
org.alljoyn.Config	1	yes	/Config

4.2 Properties

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

4.3 Methods

The following methods are exposed by the object that implements the org.alljoyn.Config interface.

4.3.1 FactoryReset

Inputs

None.

Output

None.

Description

Direct the device to disconnect from the personal AP, clear all previously configured data, and start the softAP mode.

Some devices may not support this feature. In such a case, the error org.alljoyn.Error.FeatureNotAvailable will be returned in the AllJoyn response.

4.3.2 Restart

Inputs

None.

Output

None.

Description

Restart the device.

4.3.3 SetPasscode

Inputs

Parameter name	Mandatory	Signature	List of values	Description
daemonRealm	no	s	N/A	Identifies the daemon's identity for secure access. This parameter is currently ignored by the Configuration service framework.
newPasscode	no	s	N/A	Passphrase that will be utilized for the secure Config interface.

Outputs

None.

Description

Update the passcode to be used for the org.alljoyn.Config interface which is secure. The default passcode is 000000 until it is overwritten by SetPasscode method.

4.3.4 GetConfigurations

Inputs

Parameter name	Mandatory	Signature	List of values	Description
languageTag	no	s	IETF language tags specified by RFC 5646	Language tag used to retrieve Config fields.

Outputs

Parameter name	Mandatory	Return signature	Description
configData	no	a{sv}	Returns configuration fields in the form of dictionary. See *Configuration map fields* for default set of Configuration map fields.

Description

Return all the configurable fields specified within the scope of the Config interface.

Error handling regarding the input parameter:

- If language tag is not specified (i.e., ""), configuration fields based on device's default language are returned.
- If a language tag is not supported by the device, Alljoyn error org.alljoyn.Error.LanguageNotSupported is returned.

4.3.5 UpdateConfigurations

Inputs

Parameter name	Mandatory	Signature	List of values	Description
languageTag	no	s	IETF language tags specified by RFC 5646	Identifies the language tag
configMap	no	a{sv}	See *Configuration map fields*	Set of configuration fields being updated.

Outputs

None.

Description

Provide a mechanism to update the configuration fields.

- Whenever there is an error in updating the value for a specific field in the configMap, the error org.alljoyn.Error.InvalidValue will be returned. The error message will contain the field name of the invalid field.
- If a language tag is not supported by the device, the error org.alljoyn.Error.LanguageNotSupported is returned.

4.3.6 ResetConfigurations

Inputs

Parameter name	Mandatory	Signature	List of values	Description
languageTag	no	s	IETF language tags specified by RFC 5646	Identifies the language tag
fieldList	no	as	N/A	List of fields or configuration items that are being reset

Outputs

None.

Description

Provide a mechanism to reset (i.e., value is restored to factory default but the field itself is retained) values of configuration fields.

- Whenever there is an error related to fieldList, the error org.alljoyn.Error.InvalidValue will be returned. The error message will contain the field name of the invalid field.
- If a language tag is not supported by the device, the error org.alljoyn.Error.LanguageNotSupported is returned.

4.3.7 Configuration map fields

Table 2 lists the known configuration fields that are part of the configMap parameter fields. The OEM or application developer can add additional fields.

Table 2: configMap parameter fields

Field name	Mandatory	Localized	Signature	Description
DefaultLanguage	yes	no	s	Default language supported by the device. IETF language tags specified by RFC 5646. ■ If the parameter is not set as per the RFC, the error org.alljoyn.Error.InvalidValue is returned. ■ If a language tag is not supported by the device, the error org.alljoyn.Error.LanguageNotSupported is returned. In this case, the default language on the device is unchanged.
DeviceName	yes	yes	s	Device name assigned by the user. The device name appears on the UI as the friendly name of the device.

4.4 Introspect XML

The following XML defines the org.alljoyn.Config interface.

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

xsi:noNamespaceSchemaLocation="http://www.allseenalliance.org/schemas/introspect.xsd">

    <interface name="org.alljoyn.Config">
        <property name="Version" type="q" access="read"/>
        <method name="FactoryReset">
          <annotation name="org.freedesktop.DBus.Method.NoReply" value="true"
/>
        </method>
        <method name="Restart">
          <annotation name="org.freedesktop.DBus.Method.NoReply" value="true"
/>
        </method>
        <method name="SetPasscode">
            <arg name="daemonRealm" type="s" direction="in"/>
            <arg name="newPasscode" type="ay" direction="in"/>
        </method>
       <method name="GetConfigurations">
            <arg name="languageTag" type="s" direction="in"/>
            <arg name="configData" type="a{sv}" direction="out"/>
        </method>
        <method name="UpdateConfigurations">
```

```
            <arg name="languageTag" type="s" direction="in"/>
            <arg name="configMap" type="a{sv}" direction="in"/>
        </method>
        <method name="ResetConfigurations">
            <arg name="languageTag" type="s" direction="in"/>
            <arg name="fieldList" type="as" direction="in"/>
        </method>
    </interface>
</node>
```