# AllJoyn™ Configuration 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™ Configuration service framework exposed by a device through the Config 1.0 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 the System App) will bundle this service framework.

**NOTE**

See Interface data to review the parameters supported by the Configuration service framework. Refer to the *AllJoyn™ Configuration 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 Config interface specifications. Successful completion of all test cases in this document does not guarantee that the tested device will interoperate with other devices.

## 1.3 Release history

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

## 1.4 References

The following are reference documents:

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

# 2 Environment Setup

## 2.1 Requirements

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

- An AllJoyn-enabled device (the device under test or DUT) that supports the AllJoyn Configuration service framework
- 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 Config interface
- The passcode for the DUT is set to the default passcode

## 2.3 Test execution

**NOTE**

Some of the test cases involve scenarios where the passcode is changed or a factory reset is requested.

If the DUT does not conform to the interface specification or if there is an issue that causes a test case step to fail, manual steps may be necessary to return the DUT to the proper state (e.g. passcode set to the default passcode) before testing can continue.

If the DUT does not support the Onboarding interface but does support FactoryReset, some test cases will prompt the tester after the factory reset call to perform any required steps to connect the DUT to the personal AP.

## 2.4 Parameters

**Table 1. Parameters for the Configuration service framework**

| Parameter | Description |
|---|---|
| DeviceId | Device ID of the DUT |
| AppId | Application ID of the System application on the DUT (application supporting the Config interface) |

# 3 Config Test Cases

## 3.1 Config-v1-01: System App AppId equals DeviceId

**Objective**

Verify whether the AppId of the DUT's System App (the application supporting the Config interface) matches the DUT's DeviceId.

**Procedure**

1. The test device waits to receive an About announcement from the application on the DUT.
2. After receiving an About Announcement, the test device joins a session with the application at the port specified in the received About announcement.
3. 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 DeviceId and AppId fields in the metaData parameter of the received About announcement match. That is, the DeviceId is equal to the string representation of the UUID for the AppId.
    - If these do not match, a note is added to this effect.
    - In either case, the test case passes.

## 3.2 Config-v1-02: Call a Config interface method without proper authentication

**Objective**

Verify that the test device cannot call methods on the DUT's Config bus object when the wrong passcode is given.

**Procedure**

1. The test device waits to receive an About announcement from the application on the DUT.
2. After receiving an About Announcement, 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 wrong passcode (“123456”) when authentication is requested.
4. The test device tries to retrieve the Version property from the Config bus object.
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 receives an error when attempting to retrieve the Version property on the Config bus object and the completed method is called on the registered AuthListener indicating that the authentication failed.

## 3.3 Config-v1-04: GetConfigurations() method with default language

**Objective**

Verify that calling the GetConfigurations() method with a languageTag parameter equal to the default language returns values consistent with the About announcement.

**Procedure**

1. The test device waits to receive an About announcement from the application on the DUT.
2. After receiving an About Announcement, 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 calls the GetConfigurations() method on the Config bus object providing the default language for the languageTag parameter (the default language is the value received for the DefaultLanguage field in the metaData parameter of the received About announcement).
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 GetConfigurations() method returns a map containing fields for the DefaultLanguage and the DeviceName.

- The DefaultLanguage and DeviceName values match those received in the metaData parameter of the received About announcement.

## 3.4 Config-v1-05: GetConfigurations() method with unspecified language

### Objective

Verify that calling the GetConfigurations() method with an unspecified language for the languageTag parameter returns the same fields and values as calling the GetConfigurations() method with a languageTag parameter equal 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, 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 calls the GetConfigurations() method on the Config bus object providing “” for the languageTag parameter.
5. The test device calls the GetConfigurations() method on the Config bus object providing the default language for the languageTag parameter (the default language is the value received for the DefaultLanguage field in the metaData parameter of the received About announcement).
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 maps returned from the GetConfigurations() method calls both contain the required fields, DefaultLanguage and DeviceName, and the same values are returned for each.

# 3.5 Config-v1-06: GetConfigurations() method for each supported language

### Objective

Verify that calling the GetConfigurations() method on the Config bus object with a languageTag parameter set to each of the supported languages returns values consistent with those returned from calling the getAboutData() method on the About bus object for the same language.

### Procedure

1. The test device waits to receive an About announcement from the application on the DUT.
2. After receiving an About Announcement, 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 calls the GetAboutData() method on the About bus object with the default language for the languageTag.
5. If the SupportedLanguages field contains more than one language, the test device performs the following steps for each language:
   a. Call the GetConfigurations() method with that language on the Config bus object.
   b. Call the GetAboutData() method with that language on the About bus object.
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.
- If the SupportedLanguages field contains only one language, a note is added to this effect.
- The fields returned from each call to GetConfigurations() and GetAboutData() include DefaultLanguage and DeviceName .
- The field values for DefaultLanguage and DeviceName returned from the GetConfigurations() method and GetAboutData() method are the same for each supported language.

## 3.6 Config-v1-07: GetConfigurations() method with unsupported language

### Objective

Verify that calling the GetConfigurations() method on the Config bus object with a languageTag parameter set to an unsupported language returns a LanguageNotSupported error.

### Procedure

1. The test device waits to receive an About announcement from the application on the DUT.
2. After receiving an About Announcement, 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 calls the GetConfigurations() method on the Config bus object with a language that is not supported—"INVALID".
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 call to the GetConfigurations() method on the Config bus object with an unsupported language returns an org.alljoyn.Error.LanguageNotSupported error.

## 3.7 Config-v1-08: UpdateConfigurations() method with a new DeviceName

### Objective

Verify that the DeviceName can be changed by calling UpdateConfigurations() with a new DeviceName.

### Procedure

1. The test device waits to receive an About announcement from the application on the DUT.
2. After receiving an About Announcement, 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 calls the UpdateConfigurations() method on the Config bus object with a languageTag parameter set to the default language and a configMap parameter containing the DeviceName field with a value of "newDeviceName".
5. The test device waits to receive an updated About announcement.
6. The test device calls the GetConfigurations() method on the Config bus object with a languageTag parameter set to the default language.
7. The test device calls the GetAboutData() method on the About bus object with a languageTag parameter set to the default language.
8. The test device calls the UpdateConfigurations() method on the Config bus object with a languageTag parameter set to the default language and a configMap parameter containing the DeviceName field and the original DeviceName value.
9. The test device waits to receive an updated About announcement.
10. The test device calls the GetConfigurations() method on the Config bus object with a languageTag parameter set to the default language.
11. The test device calls the GetAboutData() method on the About bus object with a languageTag parameter set to the default language.
12. 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.
- After calling the UpdateConfigurations() method to update the DeviceName, a new About announcement is received. The About announcement contains an updated DeviceName in the metaData parameter.
- The subsequent calls to GetConfigurations() and GetAboutData() return the updated DeviceName.
- After calling the UpdateConfigurations() method to update the DeviceName to its original value, a new About announcement is received. The About announcement contains a DeviceName with the original value in the metaData parameter.
- The subsequent calls to GetConfigurations() and GetAboutData() return the original DeviceName.

## 3.8 Config-v1-12: UpdateConfigurations() method with a DeviceName containing special characters

### Objective

Verify that calling the UpdateConfigurations() method on the Config bus object with a DeviceName containing special characters will update the DeviceName and send out an updated About announcement.

### Procedure

1. The test device waits to receive an About announcement from the application on the DUT.
2. After receiving an About Announcement, 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 calls the UpdateConfigurations() method on the Config bus object with the default language for the languageTag parameter and a configMap parameter containing the DeviceName field and a value containing special characters. The value containing special characters is a string consisting of characters with the ASCII values 33 to 47, 58 to 64, 91 to 96, and 123 to 126 (inclusive).
5. The test device waits to receive an updated About announcement from the application on the DUT.
6. The test device calls the GetConfigurations() method on the Config bus object with the default language for the languageTag parameter.
7. The test device calls the GetAboutData() method on the About bus object with the default language for the languageTag parameter.
8. The test device calls the UpdateConfigurations() method on the Config bus object with the following:
   - A languageTag parameter set to the default language.
   - A configMap parameter containing the DeviceName field and the original DeviceName value received in the metaData parameter of the initially received About announcement.
9. The test device waits to receive an updated About announcement from the application on the DUT.
10. The test device calls the GetConfigurations() method on the Config bus object with the default language for the languageTag parameter.
11. The test device calls the GetAboutData() method on the About bus object with the default language for the languageTag parameter.
12. 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 receives an updated About announcement after calling the UpdateConfigurations() method with the new DeviceName.

- The DeviceName field returned from the GetConfigurations() and GetAboutData() methods equals the new DeviceName.
- The test device receives an updated About announcement after calling the UpdateConfigurations() method with the original DeviceName.
- The DeviceName field returned from the GetConfigurations() and GetAboutData() methods equals the DeviceName received in the metaData parameter of the initially received About announcement.

## 3.9 Config-v1-13: UpdateConfigurations() method with an unsupported language

### Objective

Verify that calling the UpdateConfigurations() method on the Config bus object with an unsupported language for the languageTag parameter will return an org.alljoyn.Error.LanguageNotSupported error.

### Procedure

1. The test device waits to receive an About announcement from the application on the DUT.
2. After receiving an About Announcement, 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 calls the UpdateConfigurations() method on the Config bus object with “INVALID” for the languageTag parameter and a configMap parameter containing the DeviceName field and the DeviceName value received in the About announcement.
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 call to the UpdateConfigurations() method on the Config bus object returns an org.alljoyn.Error.LanguageNotSupported error.

## 3.10 Config-v1-14: UpdateConfigurations() method with a DefaultLanguage of another language

**Objective**

Verify that calling the UpdateConfigurations() method on the Config bus object with a DefaultLanguage set to another supported language will update the DefaultLanguage and send out an updated About announcement.

**Procedure**

1. The test device waits to receive an About announcement from the application on the DUT.
2. After receiving an About Announcement, 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 calls the GetAboutData() method on the About bus object to retrieve the list of supported languages. If the list contains only one language, the test case exits.
5. The test device calls the UpdateConfigurations() method on the Config bus object with the following:
   - A languageTag parameter set to the default language.
   - a configMap parameter containing the DefaultLanguage field and a value set to another supported language value (the first supported language in the list that is not the current default language value).
6. The test device waits to receive an updated About announcement from the application on the DUT.
7. The test device calls the GetConfigurations() method on the Config bus object with the new default language for the languageTag parameter.
8. The test device calls the GetAboutData() method on the About bus object with the new default language for the languageTag parameter.
9. The test device calls the UpdateConfigurations() method on the Config bus object with the new default language for the languageTag parameter and a configMap parameter containing the DefaultLanguage field and the original supported language value.
10. The test device waits to receive an updated About announcement from the application on the DUT.
11. The test device calls the GetConfigurations() method on the Config bus object with the original default language for the languageTag parameter.
12. The test device calls the GetAboutData() method on the About bus object with the original default language for the languageTag parameter.

13. 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 the list of supported languages retrieved from the GetAboutData() method contains only one language, a note is added to this effect.
- The test device receives an updated About announcement after calling the UpdateConfigurations() method with the new default language.
- The subsequent calls to GetConfigurations() and GetAboutData() return the updated default language.
- After calling the UpdateConfigurations() method to update the DefaultLanguage field to its original value, a new About announcement is received. The About announcement contains a DefaultLanguage with the original value in the metaData parameter.
- The subsequent calls to GetConfigurations() and GetAboutData() return the original default language.

## 3.11 Config-v1-15: UpdateConfigurations() method for DefaultLanguage with an unsupported language

**Objective**

Verify that calling the UpdateConfigurations() method on the Config bus object with a DefaultLanguage set to an unsupported language will return an org.alljoyn.Error.LanguageNotSupported error.

**Procedure**

1. The test device waits to receive an About announcement from the application on the DUT.
2. After receiving an About Announcement, 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 calls the UpdateConfigurations() method on the Config bus object with the default language for the languageTag parameter and a configMap parameter containing the DefaultLanguage field and a value set to “INVALID”.
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 call to the UpdateConfigurations() method on the Config bus object returns an org.alljoyn.Error.LanguageNotSupported error.

## 3.12 Config-v1-16: UpdateConfigurations() method for DefaultLanguage with an unspecified language

### Objective

Verify that calling the UpdateConfigurations() method on the Config bus object with a DefaultLanguage set to an empty string will return an org.alljoyn.Error.LanguageNotSupported error.

### Procedure

1. The test device waits to receive an About announcement from the application on the DUT.
2. After receiving an About Announcement, 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 calls the UpdateConfigurations() method on the Config bus object with the default language for the languageTag parameter and a configMap parameter containing the DefaultLanguage field and a value set to “”.
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 call to the UpdateConfigurations() method on the Config bus object returns an org.alljoyn.Error.LanguageNotSupported error.

## 3.13 Config-v1-19: UpdateConfigurations() method for an invalid field

### Objective

Verify that calling the UpdateConfigurations() method on the Config bus object with an invalid field will return an org.alljoyn.Error.InvalidValue error.

### Procedure

1. The test device waits to receive an About announcement from the application on the DUT.
2. After receiving an About Announcement, 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 calls the UpdateConfigurations() method on the Config bus object with the default language for the languageTag parameter and a configMap parameter containing the field "INVALID" and a value of "INVALID".
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 call to the UpdateConfigurations() method on the Config bus object returns an org.alljoyn.Error.InvalidValue error.

## 3.14 Config-v1-20: ResetConfigurations() method for DeviceName

### Objective

Verify that calling the ResetConfigurations() method on the Config bus object with the DeviceName will reset the DeviceName to its default value.

### Procedure

1. The test device waits to receive an About announcement from the application on the DUT.
2. After receiving an About Announcement, 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 calls the ResetConfigurations() method on the Config bus object with a languageTag parameter set to the default language and the fieldList parameter consisting of the DeviceName field.
5. The test device calls the GetConfigurations() method on the Config bus object with the default language for the languageTag parameter.
6. The test device calls the GetAboutData() method on the About bus object with the default language for the languageTag parameter.

7. The test device calls the UpdateConfigurations() method on the Config bus object with the languageTag parameter set to the default language and a configMap parameter containing the DeviceName field with a value of "newDeviceName".
8. The test device waits to receive an updated About announcement.
9. The test device calls the GetConfigurations() method on the Config bus object with a languageTag parameter set to the default language.
10. The test device calls the GetAboutData() method on the About bus object with a languageTag parameter set to the default language.
11. The test device calls the ResetConfigurations() method on the Config bus object with the languageTag parameter set to the default language and the fieldList parameter consisting of the DeviceName field.
12. The test device waits to receive an updated About announcement.
13. The test device calls the GetConfigurations() method on the Config bus object with a languageTag parameter set to the default language.
14. The test device calls the GetAboutData() method on the About bus object with a languageTag parameter set to the default language.
15. 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.
- After calling the ResetConfigurations() method, calls to GetConfigurations() and GetAboutData() return the same default DeviceName value.
- After calling the UpdateConfigurations() method to update the DeviceName, a new About announcement is received. The About announcement contains an updated DeviceName in the metaData parameter. Subsequent calls to GetConfigurations() and GetAboutData() return the updated DeviceName value.
- After making a subsequent call to the ResetConfigurations() method, a new About announcement is received. The About announcement contains the same default DeviceName value seen earlier. Subsequent calls to GetConfigurations() and GetAboutData() return the same default DeviceName value.

## 3.15 Config-v1-21: ResetConfigurations() method for DefaultLanguage (at least one supported language)

### Objective

Verify that calling the ResetConfigurations() method on the Config bus object with the DefaultLanguage will reset the DefaultLanguage to its default value.

### Procedure

1. The test device waits to receive an About announcement from the application on the DUT.
2. After receiving an About Announcement, 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 calls the ResetConfigurations() method on the Config bus object with a languageTag parameter set to the default language and the fieldList parameter consisting of the DefaultLanguage field.
5. The test device calls the GetConfigurations() method on the Config bus object with the default language for the languageTag parameter.
6. The test device calls the GetAboutData() method on the About bus object with the default language for the languageTag parameter.
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.
- After calling the ResetConfigurations() method, calls to GetConfigurations() and GetAboutData() return the same default DefaultLanguage value.

## 3.16 Config-v1-22: ResetConfigurations() method for DefaultLanguage (more than one supported language)

### Objective

Verify that calling the ResetConfigurations() method on the Config bus object with the DefaultLanguage will reset the DefaultLanguage to its default value after having called the UpdateConfigurations() method on the Config bus object to change the DefaultLanguage to another supported language.

### Procedure

1. The test device waits to receive an About announcement from the application on the DUT.
2. After receiving an About Announcement, 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 calls the GetAboutData() method on the About bus object to retrieve the list of supported languages. If the list contains only one language, the test case exits.
5. The test device calls the ResetConfigurations() method on the Config bus object with the languageTag parameter set to the default language and the fieldList parameter consisting of the DefaultLanguage field.
6. The test device calls the GetConfigurations() method on the Config bus object with "" for the languageTag parameter.
7. The test device calls the GetAboutData() method on the About bus object with "" for the languageTag parameter.
8. The test device calls the UpdateConfigurations() method on the Config bus object with the following:
   - A languageTag parameter set to the default language.
   - A configMap parameter containing the DefaultLanguage field and a value set to another supported language value (the first supported language in the list that is not the current default language value).
9. The test device waits to receive an updated About announcement from the application on the DUT.
10. The test device calls the GetConfigurations() method on the Config bus object with "" for the languageTag parameter.
11. The test device calls the GetAboutData() method on the About bus object with "" for the languageTag parameter.
12. The test device calls the ResetConfigurations() method on the Config bus object with a languageTag parameter set to the current default language and the fieldList parameter consisting of the DefaultLanguage field.
13. The test device waits to receive an updated About announcement from the application on the DUT.
14. The test device calls the GetConfigurations() method on the Config bus object with "" for the languageTag parameter.
15. The test device calls the GetAboutData() method on the About bus object with "" for the languageTag parameter.
16. 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 the list of supported languages retrieved from the GetAboutData() method contains only one language, a note is added to this effect.

- After calling the ResetConfigurations() method, calls to GetConfigurations() and GetAboutData() return the same default DefaultLanguage value.
- After calling the UpdateConfigurations() method to update the DefaultLanguage, a new About announcement is received. The About announcement contains an updated DefaultLanguage in the metaData parameter. Subsequent calls to GetConfigurations() and GetAboutData() return the updated DefaultLanguage value.
- After the subsequent call to the ResetConfigurations() method, a new About announcement is received. The About announcement contains the default DefaultLanguage in the metaData parameter. Subsequent calls to GetConfigurations() and GetAboutData() return the same default DefaultLanguage value.

## 3.17 Config-v1-24: ResetConfigurations() method with an unsupported language

**Objective**

Verify that calling the ResetConfigurations() method on the Config bus object with an unsupported language for the languageTag returns an org.alljoyn.Error.LanguageNotSupported error.

**Procedure**

1. The test device waits to receive an About announcement from the application on the DUT.
2. After receiving an About Announcement, 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 calls the ResetConfigurations() method on the Config bus object with the languageTag parameter set to “INVALID” and the fieldList parameter consisting of the DeviceName field.
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 call to the ResetConfigurations() method on the Config bus object returns an org.alljoyn.Error.LanguageNotSupported error.

## 3.18 Config-v1-25: ResetConfigurations() method for an invalid field

### Objective

Verify that calling the ResetConfigurations() method on the Config bus object with an invalid field will return an org.alljoyn.Error.InvalidValue error.

### Procedure

1. The test device waits to receive an About announcement from the application on the DUT.
2. After receiving an About Announcement, 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 calls the ResetConfigurations() method on the Config bus object with the languageTag parameter set to the default language and the fieldList parameter consisting of an "INVALID" field.
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 call to the ResetConfigurations() method on the Config bus object returns an org.alljoyn.Error.InvalidValue error.

## 3.19 Config-v1-26: Restart() method

### Objective

Verify that calling the Restart() method on the Config bus object will restart the device.

### Procedure

1. The test device waits to receive an About announcement from the application on the DUT.
2. After receiving an About Announcement, 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 calls the Restart() method on the Config bus object.
5. The test device loses the session.

6. The test device waits for the device announcement and rejoins an AllJoyn session with the DUT at the port specified in the About announcement.

**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.
- After calling the Restart() method, the test device loses the session.
- The test device receives another About announcement from the application on the DUT after it restarts.
- The test device joins a session again with the application at the port specified in the received About announcement.

## 3.20 Config-v1-27: Restart() method persists configuration changes

**Objective**

Verify that calling the Restart() method on the Config bus object will restart the device and that configuration changes will be retained.

**Procedure**

1. The test device waits to receive an About announcement from the application on the DUT.
2. After receiving an About Announcement, 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 calls the UpdateConfigurations() method on the Config bus object with a languageTag parameter set to the default language and a configMap parameter containing the DeviceName field with a value of "newDeviceName".
5. The test device waits to receive an updated About announcement.
6. The test device calls the GetConfigurations() method on the Config bus object with a languageTag parameter set to the default language.
7. The test device calls the GetAboutData() method on the About bus object with a languageTag parameter set to the default language.
8. The test device calls the Restart() method on the Config bus object.
9. The test device loses the session.
10. The test device waits to receive another About announcement from the application on the DUT.

11. After receiving an About Announcement, the test device joins a session with the application at the port specified in the received About announcement.
12. The test device calls the GetConfigurations() method on the Config bus object with a languageTag parameter set to the default language.
13. The test device calls the GetAboutData() method on the About bus object with a languageTag parameter set to the default language.
14. The test device calls the UpdateConfigurations() method on the Config bus object with a languageTag parameter set to the default language and a configMap parameter containing the DeviceName field and the original DeviceName value.
15. The test device waits to receive another About announcement from the application on the DUT.
16. The test device calls the GetConfigurations() method on the Config bus object with a languageTag parameter set to the default language.
17. The test device calls the GetAboutData() method on the About bus object with a languageTag parameter set to the default language.
18. 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.
- After calling the UpdateConfigurations() method to update the DeviceName, a new About announcement is received. The About announcement contains an updated DeviceName in the metaData parameter.
- The subsequent calls to GetConfigurations() and GetAboutData() return the updated DeviceName.
- After calling the Restart() method, the test device loses the session.
- The test device receives another About announcement from the application on the DUT after it restarts.
- The test device joins a session again with the application at the port specified in the received About announcement.
- The subsequent calls to GetConfigurations() and GetAboutData() return the updated DeviceName.
- After calling the UpdateConfigurations() method to change the DeviceName back to the original value, a new About announcement is received. The About announcement contains the original value for the DeviceName in the metaData parameter.
- The subsequent calls to GetConfigurations() and GetAboutData() return the original value for the DeviceName.

# 3.21 Config-v1-29: SetPasscode() method with a new value

### Objective

Verify that calling the SetPasscode() method on the Config bus object with a newPasscode parameter set to a new passcode will change the passcode to the new value.

### Procedure

1. The test device waits to receive an About announcement from the application on the DUT.
2. After receiving an About Announcement, 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 devices calls the SetPasscode() method on the Config bus object with the daemonRealm parameter set to an empty string and the newPasscode parameter set to “111111”.
5. The test device leaves the session.
6. The test device clears its key store of authentication keys.
7. The test device joins a session with the application at the port specified in the received About announcement.
8. The test device registers an AuthListener with the AllJoyn framework that provides the new passcode (“111111”) when authentication is requested.
9. The test device retrieves the Version property from the Config bus object.
10. The test devices calls the SetPasscode() method on the Config bus object with the daemonRealm parameter set to an empty string and the newPasscode parameter set to “000000”.
11. The test device leaves the session.
12. The test device clears its key store of authentication keys.
13. The test device joins a session with the application at the port specified in the received About announcement.
14. The test device registers an AuthListener with the AllJoyn framework that provides the default passcode (“000000”) when authentication is requested.
15. The test device retrieves the Version property from the Config bus object.
16. 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.
- After calling the SetPasscode() method on the Config bus object and leaving the session, the test device can join another session with the application and retrieve the Version property.
- After calling the SetPasscode() method on the Config bus object to set the passcode back to the default value, the test device can join another session with the application and retrieve the Version property.

## 3.22 Config-v1-30: SetPasscode() method with a one-character value

**Objective**

Verify that calling the SetPasscode() method on the Config bus object with a newPasscode parameter set to a one character array will change the passcode to the new value.

**Procedure**

1. The test device waits to receive an About announcement from the application on the DUT.
2. After receiving an About Announcement, 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 devices calls the SetPasscode() method on the Config bus object with the daemonRealm parameter set to an empty string and the newPasscode parameter set to a new passcode of “1”.
5. The test device leaves the session.
6. The test device clears its key store of authentication keys.
7. The test device joins a session with the application at the port specified in the received About announcement.
8. The test device registers an AuthListener with the AllJoyn framework that provides the new passcode (“1”) when authentication is requested.
9. The test device retrieves the Version property from the Config bus object.
10. The test devices calls the SetPasscode() method on the Config bus object with the daemonRealm parameter set to an empty string and the newPasscode parameter set to the default passcode (“000000”).
11. The test device leaves the session.
12. The test device clears its key store of authentication keys.

13. The test device joins a session with the application at the port specified in the received About announcement.
14. The test device retrieves the Version property from the Config bus object.
15. 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.
- After calling the SetPasscode() method on the Config bus object and leaving the session, the test device can join another session with the application and retrieve the Version property.
- After calling the SetPasscode() method on the Config bus object to set the passcode back to the default value, the test device can join another session with the application and retrieve the Version property.

## 3.23 Config-v1-31: SetPasscode() method with special characters

**Objective**

Verify that calling the SetPasscode() method on the Config bus object with a newPasscode parameter that includes special characters will change the passcode to the new value.

**Procedure**

1. The test device waits to receive an About announcement from the application on the DUT.
2. After receiving an About Announcement, 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 devices calls the SetPasscode() method on the Config bus object with the daemonRealm parameter set to an empty string and the newPasscode parameter set to a new passcode of “!@#$%^”.
5. The test device leaves the session.
6. The test device clears its key store of authentication keys.
7. The test device joins a session with the application at the port specified in the received About announcement.
8. The test device registers an AuthListener with the AllJoyn framework that provides the new passcode (“!@#$%^”) when authentication is requested.

9. The test device retrieves the Version property from the Config bus object.
10. The test devices calls the SetPasscode() method on the Config bus object with the daemonRealm parameter set to an empty string and the newPasscode parameter set to the default passcode (“000000”).
11. The test device leaves the session.
12. The test device clears its key store of authentication keys.
13. The test device joins a session with the application at the port specified in the received About announcement.
14. The test device retrieves the Version property from the Config bus object.
15. 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.
- After calling the SetPasscode() method on the Config bus object and leaving the session, the test device can join another session with the application and retrieve the Version property.
- After calling the SetPasscode() method on the Config bus object to set the passcode back to the default value, the test device can join another session with the application and retrieve the Version property.

## 3.24 Config-v1-32: Restart() method persists changed passcode

### Objective

Verify that calling the SetPasscode() method on the Config bus object with a newPasscode parameter set to “111111” and then calling the Restart() method will require a subsequent session to use the newPasscode to retrieve the Version property.

### Procedure

1. The test device waits to receive an About announcement from the application on the DUT.
2. After receiving an About Announcement, 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 devices calls the SetPasscode() method on the Config bus object with the daemonRealm parameter set to an empty string and the newPasscode parameter set to a new passcode of “111111”.

5. The test device leaves the session.
6. The test device clears its key store of authentication keys.
7. The test device joins a session with the application at the port specified in the received About announcement.
8. The test device registers an AuthListener with the AllJoyn framework that provides the new passcode (“111111”) when authentication is requested.
9. The test device retrieves the Version property from the Config bus object.
10. The test device calls the Restart() method on the Config bus object.
11. The test device losses the session.
12. The test device clears its key store of authentication keys.
13. The test device joins a session with the application at the port specified in the received About announcement.
14. The test device retrieves the Version property from the Config bus object.
15. The test devices calls the SetPasscode() method on the Config bus object with the daemonRealm parameter set to an empty string and the newPasscode parameter set to the default passcode (“000000”).
16. The test device leaves the session.
17. The test device joins a session with the application at the port specified in the received About announcement.
18. The test device registers an AuthListener with the AllJoyn framework that provides the new passcode (“000000”) when authentication is requested.
19. The test device retrieves the Version property from the Config bus object.
20. 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.
- After calling the SetPasscode() method on the Config bus object and leaving the session, the test device can join another session with the application and retrieve the Version property.
- After calling the Restart() method, the test device loses the session.
- The test device receives another About announcement from the application on the DUT after it restarts.
- The test device joins a session again with the application at the port specified in the received About announcement.
- The test device is able to retrieve the Version property.

- After calling the SetPasscode() method on the Config bus object to set the passcode back to the default value, the test device can join another session with the application and retrieve the Version property.

## 3.25 Config-v1-33: FactoryReset() method

### Objective

Verify that calling the FactoryReset() method on the Config bus object will either perform a factory reset or return a FeatureNotAvailable error.

### Procedure

1. The test device waits to receive an About announcement from the application on the DUT.
2. After receiving an About Announcement, 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. If the About announcement advertises support of the Onboarding interface, the test case exits.
5. The test device calls the ResetConfigurations() method on the Config bus object with the languageTag parameter set to the default language and the fieldList parameter consisting of the DeviceName and DefaultLanguage field.
6. The test device calls the GetConfigurations() method on the Config bus object with “” for the languageTag parameter.
7. The test device calls the FactoryReset() method on the Config bus object.
8. If the FactoryReset() method returns an org.alljoyn.Error.FeatureNotAvailable error, the text case exits.
9. The test device waits to lose the session and then displays a message to the user that the device has been factory reset and it needs to be onboarded.
10. The user taps Continue once the device has been onboarded.
11. The test device waits to receive an About announcement from the application on the DUT.
12. After receiving an About Announcement, the test device joins a session with the application at the port specified in the received About announcement.
13. The test device calls the GetConfigurations() method on the Config bus object with “” for the languageTag parameter.
14. 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 the About announcement advertises support of the Onboarding interface, the test case adds a note to this effect.
- If the FactoryReset() method returns an org.alljoyn.Error.FeatureNotAvailable error, a note is added to this effect.
- After calling the FactoryReset() method on the Config bus object, the test device loses the session.
- After the user taps Continue, the test device receives an About announcement and joins another session with the application.
- A subsequent call to GetConfigurations() returns the same values for the DeviceName and DefaultLanguage fields as what was retrieved before the FactoryReset() call.

## 3.26 Config-v1-34: FactoryReset() method clears configured data

### Objective

Verify that calling the FactoryReset() method on the Config bus object will either perform a factory reset and clear all configured values or return a FeatureNotAvailable error.

### Procedure

1. The test device waits to receive an About announcement from the application on the DUT.
2. After receiving an About Announcement, 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. If the About announcement advertises support of the Onboarding interface, the test case exits.
5. The test device calls the ResetConfigurations() method on the Config bus object with the languageTag parameter set to the default language and the fieldList parameter consisting of the DeviceName field.
6. The test device calls the GetConfigurations() method on the Config bus object with “” for the languageTag parameter.
7. The test device calls the GetAboutData() method on the About bus object with “” for the languageTag parameter.
8. The test device calls the UpdateConfigurations() method on the Config bus object with the languageTag parameter set to the default language and a configMap parameter containing the DeviceName field with a value of "newDeviceName".

9. The test device waits to receive an updated About announcement.
10. The test device calls the GetConfigurations() method on the Config bus object with “” for the languageTag parameter.
11. The test device calls the GetAboutData() method on the About bus object with “” for the languageTag parameter.
12. The test device calls the FactoryReset() method on the Config bus object.
13. If the FactoryReset() method returns an org.alljoyn.Error.FeatureNotAvailable error, the text case exits.
14. The test device waits to lose the session and then displays a message to the user that the device has been factory reset and it needs to be onboarded.
15. The user taps Continue once the device has been onboarded.
16. The test device waits to receive an About announcement from the application on the DUT.
17. After receiving an About Announcement, the test device joins a session with the application at the port specified in the received About announcement.
18. The test device calls the GetConfigurations() method on the Config bus object with “” for the languageTag parameter.
19. The test device calls the GetAboutData() method on the About bus object with “” for the languageTag parameter.
20. 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 the About announcement advertises support of the Onboarding interface, the test case adds a note to this effect.
- After calling the ResetConfigurations() method, calls to GetConfigurations() and GetAboutData() return the same default DeviceName value.
- After calling the UpdateConfigurations() method to update the DeviceName, a new About announcement is received. The About announcement contains an updated DeviceName in the metaData parameter. Subsequent calls to GetConfigurations() and GetAboutData() return the updated DeviceName value.
- If the FactoryReset() method returns an org.alljoyn.Error.FeatureNotAvailable error, a note is added to this effect.
- After calling the FactoryReset() method on the Config bus object, the test device loses the session.
- After the user taps Continue, the test device receives an About announcement and joins another session with the application.

- Subsequent calls to GetConfigurations() and GetAboutData() return the same default DeviceName value.

## 3.27 Config-v1-35: FactoryReset() method resets the passcode

**Objective**

Verify that calling the FactoryReset() method on the Config bus object will either perform a factory reset and reset the passcode to the default value or return an org.alljoyn.Error.FeatureNotAvailable error.

**Procedure**

1. The test device waits to receive an About announcement from the application on the DUT.
2. After receiving an About Announcement, 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. If the About announcement advertises support of the Onboarding interface, the test case exits.
5. The test devices calls the SetPasscode() method on the Config bus object with the daemonRealm parameter set to an empty string and the newPasscode parameter set to “111111”.
6. The test device leaves the session.
7. The test device clears its key store of authentication keys.
8. The test device joins a session with the application at the port specified in the received About announcement.
9. The test device registers an AuthListener with the AllJoyn framework that provides the new passcode (“111111”) when authentication is requested.
10. The test device retrieves the Version property from the Config bus object.
11. The test device calls the FactoryReset() method on the Config bus object.
12. If the FactoryReset() method returns an org.alljoyn.Error.FeatureNotAvailable error, the text case exits.
13. The test device waits to lose the session and then displays a message to the user that the device has been factory reset and it needs to be onboarded.
14. The user taps Continue once the device has been onboarded.
15. The test device waits to receive an About announcement from the application on the DUT.

16. After receiving an About Announcement, the test device joins a session with the application at the port specified in the received About announcement.
17. The test device registers an AuthListener with the AllJoyn framework that provides the default passcode (“000000”) when authentication is requested.
18. The test device retrieves the Version property from the Config bus object.
19. 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 the About announcement advertises support of the Onboarding interface, the test case adds a note to this effect.
- After calling the SetPasscode() method on the Config bus object and leaving the session, the test device can join another session with the application and retrieve the Version property.
- If the FactoryReset() method returns an org.alljoyn.Error.FeatureNotAvailable error, a note is added to this effect.
- After calling the FactoryReset() method on the Config bus object, the test device loses the session.
- After the user taps Continue, the test device receives an About announcement and joins another session with the application.
- The test device can retrieve the Version property from the Config bus object.

# 4 Interface Data

This chapter lists the interface data supported by the Config interface.

## 4.1 FactoryReset method

While this method has no input or output parameters, it is considered mandatory.

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

## 4.2 Restart method

While this method has no input or output parameters, it is considered mandatory.

## 4.3 SetPasscode method

| Parameter | Mandatory |
|---|---|
| daemonRealm | no |
| newPasscode | no |

## 4.4 GetConfigurations method

See configMap parameter fields for the default set of Configuration map fields.

| Parameter | Mandatory |
|---|---|
| languageTag | no |
| configData | no |

## 4.5 UpdateConfigurations method

See configMap parameter fields for the default set of Configuration map fields related to the configMap parameter.

| Parameter | Mandatory |
|---|---|
| languageTag | no |
| configMap | no |

## 4.6 ResetConfigurations method

| Parameter | Mandatory |
|---|---|
| languageTag | no |
| fieldList | no |

## 4.7 configMap parameter fields

| Parameter | Mandatory |
|---|---|
| DefaultLanguage | yes |
| DeviceName | yes |