XXXX Service Framework Interface Definition Template

Month Day, Year

Contents

[1 Introduction 4](#_Toc383602937)

[1.1 Purpose 4](#_Toc383602938)

[1.2 Scope 4](#_Toc383602939)

[1.3 Release history 4](#_Toc383602940)

[1.4 References 5](#_Toc383602941)

[2 Specification 6](#_Toc383602942)

[2.1 Overview 6](#_Toc383602943)

[2.2 Discovery 6](#_Toc383602944)

[2.3 Discovery call flows 6](#_Toc383602945)

[2.3.1 Typical discovery flow 6](#_Toc383602946)

[2.3.2 Nontypical discovery flow 7](#_Toc383602947)

[2.4 Error handling 8](#_Toc383602948)

[2.5 XXXX interface 8](#_Toc383602949)

[2.5.1 Interface name 9](#_Toc383602950)

[2.5.2 Properties 9](#_Toc383602951)

[2.5.3 Methods 9](#_Toc383602952)

[2.5.4 Signals 10](#_Toc383602953)

[2.5.5 AllJoyn Introspection XML 10](#_Toc383602954)

[2.6 XXXX interface 11](#_Toc383602955)

[2.6.1 Interface name 11](#_Toc383602956)

[2.6.2 Properties 11](#_Toc383602957)

[2.6.3 Methods 11](#_Toc383602958)

[2.6.4 Signals 13](#_Toc383602959)

[2.6.5 AllJoyn Introspection XML 13](#_Toc383602960)

Figures

[Figure 1. Typical discovery call flow (client discovers a service app) 7](#_Toc383602961)

[Figure 2. Nontypical discovery call flow 8](#_Toc383602962)

Tables

[Table 1. XXXX service interface errors 8](#_Toc383602963)

[Table 2. XXXX interface data fields 9](#_Toc383602964)

[Table 3. XXXX interface data fields 12](#_Toc383602965)

# Introduction

[XXXX in the document title refers the service framework name. In the body of this document, if your service framework supports one interface, all XXXXs refer to that interface. If there is more than one, you can refer to them as the XXXX service framework’s collection of interfaces.

***Version Numbering:***

* *Core dependency: <core-year>.<core-month>, e.g. 14.06*
* *Feature release level: For use by services with more than one feature release built on the same core version*
* *Bug fix: versions not affecting incompatibility and certification*

*Append a letter for bug fix releases:*

*< core-year>.<core-month>.<feature><bugfix-letter>*

*Examples:*

* *14.06.00 is an initial service release using core 14.06*
* *14.06.00a is a bug fix release*
* *14.06.00b is another bug fix release*
* *14.06.01 is the next service feature release using core 14.06*

*As before, bug fix releases cannot add features, modify APIs, or introduce incompatible behavior. They are only intended to address critical stability and security issues.*

*The AllJoyn core will only add features on major releases 14.12, 15.04, etc.*

*Documentation should follow this same versioning.*

Red bracketed text is provided throughout the document to help create the document. Delete this text before contributing the document to the AllSeen Alliance.]

## Purpose

[The following statement is an example. Paragraph content such as the statement below uses the body style.]

This document describes the specification of the XXXX interface. This interface is required by an application to [provide a definition of what the service framework does].

## Scope

[The following is a blanket statement. Add more to the section as needed.]

This document is targeted to the developers for AllJoyn applications.

## Release history

[Identify the release version in which your service framework is being contributed. The format used is <YY.MM>, for example 14.06. For subsequent release versions where updates are made, list the interfaces that were added or updated. The table provides a sample of what to include.

The table uses the table heading style for the heading. The table content uses the table entry style. Bullet points use the table bullet lvl1 style.]

|  |  |
| --- | --- |
| Release version | What changed |
| <YY.MM> | The XXXX service framework's collection of interfaces were added:   * XXXX interface version 1 * XXXX interface version 1 * XXXX interface version 1 |
| <YY.MM> | The following interfaces were added:   * XXXX interface version 1 * XXXX interfaces version 1 |

## References

[The documents below are required. If your service depends on other specifications or RFCs, please list them by the RFC title and create a hyperlink to the RFC page.]

The following are reference documents found on the AllSeen Alliance web site's Docs and Downloads section.

* Introduction to the AllJoyn Framework
* Introduction to AllJoyn Thin Library

# Specification

## Overview

[Provide information that helps explain the service framework and any helpful information about the interface(s). Typically, a diagram that illustrates the relationship between a server app and client app is also provided. Note that any diagrams you contribute to the AllSeen Alliance become property of the Alliance.

Apply the anchor style to the diagram, and insert a figure caption below the diagram.]

## Discovery

[Below is a blanket statement used to define the discovery feature. Include anything relevant for your service framework as needed. If your service framework's interface doesn't use discovery, delete this section.]

A client can discover the app via an announcement which is a sessionless signal containing the 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 client to determine whether the app provides functionality of interest.

## Discovery call flows

[Based on the answer above, if your interfaces don't use the discovery feature, delete this section.]

### Typical discovery flow

Figure 1 illustrates a typical call flow for a client to discover a service app. The client merely relies on the sessionless announcement to decide whether to connect to the service app to use its service offering.



Figure . Typical discovery call flow (client discovers a service app)

### Nontypical discovery flow

Figure 2 illustrates a call flow for a client to discover a service app and make a request for more detailed information.



Figure . Nontypical discovery call flow

## Error handling

[Delete this section if error handling does not apply. The table shows an example of an error name.]

The method calls in the XXXX interface will 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 XXXX interface.

Table . XXXX service interface errors

| Error name | Error message |
| --- | --- |
| org.alljoyn.Error.LanguageNotSupported | Language not supported |

## XXXX interface

[The following tables provide a starting point for documenting an interface.]

### Interface name

|  |  |  |
| --- | --- | --- |
| Interface name | Version | Secured |
| org.alljoyn.XXXX | 1 | no |

### Properties

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

### Methods

The following methods are exposed by a BusObject that implements the org.alljoyn.XXXX interface.

#### [Method name]

[If a method has no inputs, delete the following table and type None. For the Signature column, identify the letter representing the signal (such as s for String).

If there is no output, delete the table under the Output title and type None.]

Inputs

| Parameter name | Signature | List of values | Description |
| --- | --- | --- | --- |
|  |  |  |  |

Output

| Return signature | Description |
| --- | --- |
|  |  |

Description

[Provide a description of the method.]

Field information

[This section may or may not apply to your interface. Move to another section if the services supports multiple interfaces as needed, or delete.]

Table 2 lists the names of the metadata fields. The fields with a yes value in the Announced column will also be published via the Announce signal. See Signals for more information.

Table . XXXX interface data fields

| Field name | Required | Announced | Localized | Signature | Description |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

#### [Method name]

Inputs

| Parameter name | Signature | List of values | Description |
| --- | --- | --- | --- |
|  |  |  |  |

Output

| Return signature | Description |
| --- | --- |
|  |  |

Description

TBD.

### Signals

[If your interface doesn't include signals, delete this section.

The Parameters column uses the Split Cells option for tables which can be painful if you need more columns than are listed below. ]

| Signal name | Parameters | | Sessionless | Description |
| --- | --- | --- | --- | --- |
| Announce | Name | Signature | yes | This signal is used to announce the application information and the service interfaces that it supports. The following information is provided in the signal:   * version -- Version number of the About interface. * port -- Session port. The app will listen on this port for incoming sessions. * objectDescription --Array of object paths and the list of supported interfaces provided by each object. * metaData -- Metadata. All the fields listed in Table 2 with a yes value in the Announced column are provided in this signal. |
| version | q |
| port | q |
| objectDescription | a(oas) |
| metaData | a{sv} |

### AllJoyn Introspection XML

[Replace the code in this section with your interface code. The example below is based on the About feature.]

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

<node name="/About"

[xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"](http://www.w3.org/2001/XMLSchema-instance)

[xsi:noNamespaceSchemaLocation="http://www.allseenalliance.org/schemas/introspec](http://www.alljoyn.org/schemas/introspec) t.xsd">

<interface name="org.alljoyn.About">

<property name="Version" type="q" access="read"/>

<method name="GetAboutData">

<arg name="languageTag" type="s" direction="in"/>

<arg name="aboutData" type="a{sv}" direction="out"/>

</method>

<method name="GetObjectDescription">

<arg name="objectDescription" type="a(sas)" direction="out"/>

</method>

<signal name="Announce">

<arg name="version" type="q"/>

<arg name="port" type="q"/>

<arg name="objectDescription" type="a(sas)"/>

<arg name="metaData" type="a{sv}"/>

</signal>

</interface>

</node>

## XXXX interface

[A second interface section is provided if your service framework supports multiple interfaces. Modify the information as needed, or delete the section.]

### Interface name

|  |  |  |  |
| --- | --- | --- | --- |
| Interface name | Version | Secured | Suggested object path |
| org.alljoyn.XXXX | 1 | no | /XXXX |

### Properties

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

### Methods

The following methods are exposed by a BusObject that implements the org.alljoyn.XXXX interface.

#### [Method name]

Inputs

| Parameter name | Signature | List of values | Description |
| --- | --- | --- | --- |
|  |  |  |  |

Output

| Return signature | Description |
| --- | --- |
|  |  |

Description

TBD.

Field information

Table 3 lists the names of the metadata fields. The fields with a yes value in the Announced column will also be published via the Announce signal. See Signals for more information.

Table . XXXX interface data fields

| Field name | Required | Announced | Localized | Signature | Description |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

#### [Method name]

Inputs

| Parameter name | Signature | List of values | Description |
| --- | --- | --- | --- |
|  |  |  |  |

Output

| Return signature | Description |
| --- | --- |
|  |  |

Description

TBD.

### Signals

| Signal name | Parameters | | Sessionless | Description |
| --- | --- | --- | --- | --- |
| Announce | Name | Signature | yes | This signal is used to announce the application information and the service interfaces that it supports. The following information is provided in the signal:   * version -- Version number of the About interface. * port -- Session port. The app will listen on this port for incoming sessions. * objectDescription --Array of object paths and the list of supported interfaces provided by each object. * metaData -- Metadata. All the fields listed in Table 2 with a yes value in the Announced column are provided in this signal. |
| version | q |

### AllJoyn Introspection XML

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

<node name="/About"

[xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"](http://www.w3.org/2001/XMLSchema-instance)

[xsi:noNamespaceSchemaLocation="http://www.allseenalliance.org/schemas/introspec](http://www.alljoyn.org/schemas/introspec) t.xsd">

<interface name="org.alljoyn.About">

<property name="Version" type="q" access="read"/>

<method name="GetAboutData">

<arg name="languageTag" type="s" direction="in"/>

<arg name="aboutData" type="a{sv}" direction="out"/>

</method>

<method name="GetObjectDescription">

<arg name="objectDescription" type="a(sas)" direction="out"/>

</method>

<signal name="Announce">

<arg name="version" type="q"/>

<arg name="port" type="q"/>

<arg name="objectDescription" type="a(sas)"/>

<arg name="metaData" type="a{sv}"/>

</signal>

</interface>

</node>