ALLSEEN ALLIANCE

# *Getting Started with the AllJoyn™ Lighting Service Framework 14.06*

## Lighting SDK Android

***September 30, 2014***

# Contents

# 1 Introduction

The AllJoyn™ Lighting SDK provides a simple Lighting API to application developers to build AllJoyn™ applications to control the Lighting solution provided by Lighting service framework.

The Lighting service framework comprises two service components:

- Lamp service: Enables an embedded lighting device (such as a connected light bulb) to be controlled by the Controller service.
- Controller service: Enables AllJoyn™ applications (e.g., an application running on a smartphone) to control the Lamp service.

Lighting OEMs can embed the Lamp service in the firmware of their lighting products to enable Smart Lighting features.

The Controller service is designed to find all devices running the Lamp service on the AllJoyn™ network and provide additional functionality to control the lighting devices in a variety of ways (e.g., create and control a group of lights simultaneously and applying custom lighting effects). The Controller service can reside in lighting device, a hub, router, gateway, mobile device (smartphone or tablet), desktop, or home automation controller.

Developers can build AllJoyn™ applications that will communicate with the Controller service. Lighting SDK is a thin-layer with simple APIs which abstract the Controller service client.

**Note** This documentation is for the Lighting SDK - Android.

*Figure 1* illustrates the Lighting service framework components and their interaction with an OEM device.

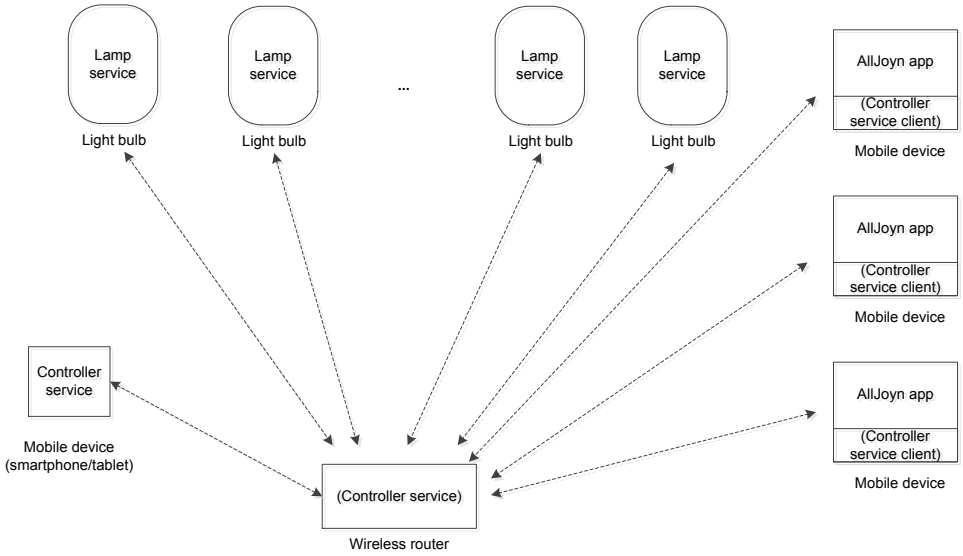


**Figure 1: Lighting service framework components**

# 1.1 Purpose

This document provides an overview of the Lighting service framework and detailed instructions on how to start using the Lighting SDK. The following topics are covered:

- Overview of the Lighting service framework
- Overview of the Lighting SDK
- Obtaining and building the Tutorial lighting apps using Lighting SDK

# 1.2 Scope

This document is written for OEMs and developers and assumes competent knowledge of AllJoyn™ development. It will focus on the main components of the Lighting service framework.

# 1.3 References

The following documents are references found on the AllSeen Alliance web site's *Docs/Downloads* section.

- *AllJoyn™™ Framework Tutorial*
- *Introduction to the AllJoyn™™ Framework*
- *Introduction to AllJoyn™™ Thin Library*
- *AllJoyn™™ Troubleshooting Guide*
- *Getting Started AllJoyn™ Lighting Service Framework – Lamp Service*
- *Getting Started AllJoyn™ Lighting Service Framework – Controller Service*

# 1.4 Acronyms and terms

| Term | Definition |
|---|---|
| AllJoyn™ Standard Core Library | An application or AllJoyn™ daemon process that contains the full implementation of the AllJoyn™ message bus. |
| Announcement | A sessionless signal whose payload includes published services' interfaces and metadata that are used for discovery. |
| Configuration service framework | Software layer that enables devices to provide remote configuration of AllJoyn™ service frameworks' metadata (ConfigData) in a session. |
| Controller service | Maintains the Lamp Group, Scene, and Presets on behalf of the Lamps. The Controller service is a logical entity that can reside in a hub, router, gateway, smartphone/tablet, desktop, or home automation controller. |

| Term | Definition |
|---|---|
| Lamp group | A logical grouping of Lamps allowing them to be controlled simultaneously as they are a single Lamp |
| Lamp service | Client service implemented in a Lamp, enabling it to be controlled by the Controller service. A Lamp service can reside in a lamp, luminaire, light switch, plug, outlet, or socket adapter and can leverage the AllJoyn™ Standard Core or Thin Client Library. |
| Lighting service framework | AllJoyn™ framework that provides a means to build a complete lighting solution. It consists of the Controller service and Lamp service. |
| Notification service framework | Software layer that enables devices to send or receive human-consumable notifications. |
| Onboarding service framework | Software layer that enables devices to provide remote configuration (OnboardingData) and control (driver mode) over a device's onboarding process to a Wi-Fi AP over an AllJoyn™ session. |
| Preset | Saving the Lamp or Lamp group state (Hue, Saturation, Color Temperature, Brightness) and giving it a friendly name. |
| Scene | Lighting setup for a particular event, for example, dimming the lights when a movie is on. The user can create a scene that they can apply to a set of lights. |
| | |
| | |
| | |

# 2 Overview

The Lighting service framework is part of a comprehensive lighting system that allows OEMs and developers to build lighting hardware that performs the following functions:

- Runs the Lamp service
- Create applications that interface with the Controller service to control lighting hardware.

## 2.1 Lamp service

The Lamp service uses the AllJoyn™ Thin Core Library and is meant to be run on embedded hardware like a light bulb. The Lamp service uses several AllJoyn™ service frameworks to allow the embedded hardware to communicate with other devices on the AllJoyn™ network.

1. The Onboarding service framework connects the hardware to the Wi-Fi network.
2. Once onboarded, the About feature allows the Lamp service to broadcast its presence on the network.
3. Any device that receives the About announcement can connect directly to the Lamp service and use the Configuration service framework to modify settings as well as perform other operations like factory reset.
4. The Lamp service uses the Notification service framework to notify the user of any problems with the light.

## 2.2 Controller service

The Controller service uses the AllJoyn™ Standard Library and is configured to listen for any lighting devices running the Lamp service that may be present on the network. The Controller service will discover any lighting device connected to the network by receiving the About announcement and then connecting directly to the device using an AllJoyn™ session. The Controller service can then perform operations on the Lamp service such as:

- Toggle the light on and off
- Retrieve power consumption and light details
- Set the brightness or hue of the light

The Controller service can control every lighting device connected to your network. Developers can then build AllJoyn™ applications that communicate directly with the Controller service and control their lighting devices. The Controller service also provides developers the ability to grouped lights together and control them simultaneously, and create a custom lighting experience using scenes. Developers have the freedom to create a variety of applications to control their lights.

In order to provide a easy way for developers to use Controller service, Controller service client was created. Controller service client provides a set of C/C++ APIs to application developers to access Controller service.

## 2.3 Lighting SDK

Lighting SDK consists of 2 components:

- Simple set of Android-native lighting APIs abstracting the Controller service client and the setup complexity of AllJoyn™
- Simple tutorial Android application which illustrates how to use the lighting SDK API

Lighting SDK APIs is a thin-layer built on top of the Controller service client. These APIs are Android-native APIs, so developers can directly use these APIs in their application without dealing with JNI.

Developers are encouraged to use the provided tutorial Android application as a reference.

*Figure 2* illustrates the Controller service, Controller service client framework, and Lighting SDK

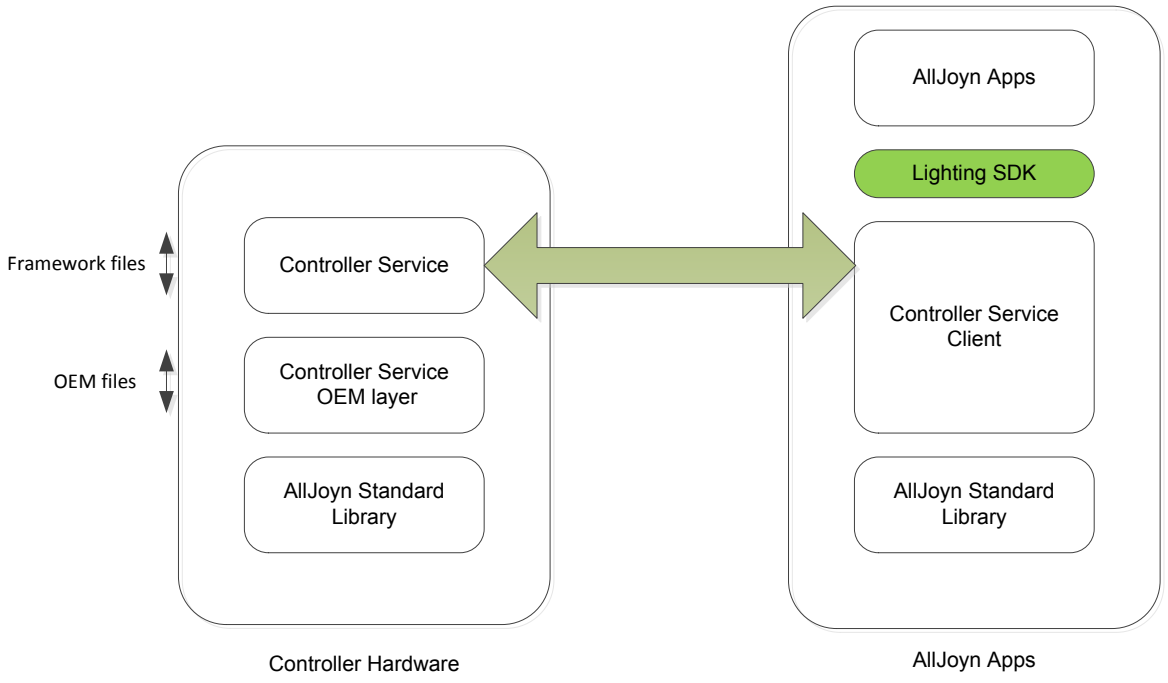


**Figure 2: Controller Service, Controller Service Client, and Lighting SDK**

# 3 Obtaining Lighting SDK

This chapter provides instructions to configure the Lighting SDK.

Verify your allseenalliance.org account has been created and you have received email confirmation before completing the procedures in this chapter.

1. Access AllSeen Alliance Lighting working group Wiki: https://wiki.allseenalliance.org/tsc/connected_lighting.
2. From the wiki, it has a direct link for downloading the Lighting SDK Android zip file.
3. In the Lighting SDK Android zip file, here is the high-level file structure:
    - `Android\README.txt`
        - Detailed step-by-step instruction on how to load and build the tutorial application on Eclipse.
    - `Android\Libraries`
        - Consists of Lighting service framework helper libraries, which abstract the Controller service client
        - Consists of AllJoyn™ libraries
    - `Android\LSFTutorial`
        - Consists of a standalone Android application illustrating how to use Lighting SDK APIs to perform simple lighting operations
        - `....\src\org\allseen\lsf\tutorial\TutorialActivity.java`
            - This is the only source file for this tutorial application.

**Note** The full source of Lighting SDK libraries is available in AllSeen Alliance.

# 4 Overview of Lighting SDK

This chapter provides an overview of the Lighting SDK

## 4.1 LightingDirector class

In the Lighting SDK, the single entry class for developers is `LightingDirector` class. LightingDirector class consists of the following major APIs:

- `start()`
  - Initializes and establishes connection with the Lighting Controller via AllJoyn™ in the network.
- `stop()`
  - Stops and cleans up the `LightingDirector` and its resources.
- `postOnNextControllerConnection()`
  - Lighting SDK will executes this callback method when a successful connection with Lighting Controller is established.
- `getLamps()`
  - Get an array of all active lamp objects (see Section 4.2) in the network
- `getGroups()`
  - Get an array of all active lamp group objects (see Section 4.3) in the network
- `getScenes()`
  - Get an array of all active scene objects (see Section 4.4) available in the network
- `getVersion()`
  - Get the API version. If there is any revisions, and additions to the APIs under `LightingDirector` class, the API version will be updated.
- `getBusAttachment()`
  - Get the AllJoyn™ Bus Attachment object created by Lighting SDK

**Note** Detailed comment and method argument are documented as JavaDocs. Please refer to https://wiki.allseenalliance.org/tsc/connected_lighting for the latest info.

**Note** Additional APIs may be added and revised in AllSeen Alliance. Please refer to https://wiki.allseenalliance.org/tsc/connected_lighting for the latest info.

## 4.2 Lamp class

In the Lighting SDK, the Lamp objects returned by `LightingDirector::getLamps()` is of type of `Lamp` class:

- `turnOn()`
  - Turn on the Lamp
- `turnOff()`
  - Turn off the Lamp
- `setColor(Hue, Saturation, Brightness, ColorTemperature)`
  - Set the Lamp state based on the provided arguments, Hue, Saturation, Brightness, and Color Temperature.

**Note** Detailed comment and method argument are documented as JavaDocs. Please refer to https://wiki.allseenalliance.org/tsc/connected_lighting for the latest info.

**Note** Additional APIs may be added and revised in AllSeen Alliance. Please refer to https://wiki.allseenalliance.org/tsc/connected_lighting for the latest info.

## 4.3 Group class

In the Lighting SDK, the Lamp group objects returned by `LightingDirector::getGroups()` is of type of `Group` class:

- `turnOn()`
  - Turn on the Lamp group
- `turnOff()`
  - Turn off the Lamp group
- `setColor(Hue, Saturation, Brightness, ColorTemperature)`
  - Set the Lamp group state based on the provided arguments, Hue, Saturation, Brightness, and Color Temperature.

**Note** Detailed comment and method argument are documented as JavaDocs. Please refer to https://wiki.allseenalliance.org/tsc/connected_lighting for the latest info.

**Note** Additional APIs may be added and revised in AllSeen Alliance. Please refer to https://wiki.allseenalliance.org/tsc/connected_lighting for the latest info.

## 4.4 Scene class

In the Lighting SDK, the Lamp group objects returned by `LightingDirector::getScenes()` is of type of `Scene` class:

- `apply()`
  - Apply the scene

**Note** Detailed comment and method argument are documented as JavaDocs. Please refer to https://wiki.allseenalliance.org/tsc/connected_lighting for the latest info.

**Note** Additional APIs may be added and revised in AllSeen Alliance. Please refer to https://wiki.allseenalliance.org/tsc/connected_lighting for the latest info.