

# **Barco Device Interface Layout Service API**

## **Barco Device Interface Layout Service API**

# Table of Contents

Glossary .....	1
Introduction .....	ii
1. Layout Service Overview .....	3
Version .....	3
Layout Services Overview .....	3
2. Layout Services API .....	4
Layout Service API to be supported by a Device to receive media .....	4
Maintaining state for the Stream .....	5
Presence Events received by Devices .....	5

## List of Examples

2.1. Setup Request .....	4
2.2. Setup Response .....	4
2.3. Teardown Request .....	5
2.4. Teardown Response .....	5
2.5. StreamStatusEvent payload sent to the central server .....	5

# Glossary

Device Interface	This represents the external interface to a device.
Device Interface API	The API accepted by the device on its external interface
Device Interface Layout Service API	The API accepted by the device on its external interface for Layout Service
Service	The device provides the following services : DeviceAdminAgent, Media, Layout, MediaStoreAgent
Handler	The code that handles a specific API message for a given service
Agent	The Handler invokes methods on the Agent for the accepted by the device on its external interface
Engine	The Agent invokes methods on the Engine that implement the services on the device
Entity	A Source or a Destination

# Introduction

This document contains the description of the device interface API for media services

# Chapter 1. Layout Service Overview

This chapter provides the version information and a brief description of the requests support by this service

## Version

Document Version = 0.0.1  
Service Version = x.x.x  
Software Version = 4.0.x.x

## Layout Services Overview

The layout service is responsible for rendering of media

The messages described here are in conformance with the AgentMessage described in the [ref\[1\]\(API-AgentMessage.pdf\)](#)

# Chapter 2. Layout Services API

This chapter provides details of the Layout Service API

## Layout Service API to be supported by a Device to receive media

These requests are typically received by a device which is capable of receiving media streams

- The device receives a (Stream)Setup Request with the streamInfo and (Stream)Teardown Request  
The streamInfo includes streamdatadoc and mediasourcedatadoc which are described under API-MediaService.pdf  
The streamdatadoc may include the streamURL in which case the device can start playing the media  
The device will signal to the source to acquire a streamURL and then start playing the media  
The response and status updates include the StreamStatus which is described under API-MediaService.pdf

### Example 2.1. Setup Request

```
<SetupRequestData>
<streamNID><!-- Unique identifier for the stream --></streamNID>
<streamdatadoc>
  <!-- MediaDestination and Profile info -->
</streamdatadoc>

<mediasourcedatadoc>
  <!-- MediaSource info -->
</mediasourcedatadoc>

</SetupRequestData>
```

### Example 2.2. Setup Response

```
<SetupResponseData>
  <StreamStatus .../>
</SetupResponseData>
```



### Example 2.3. Teardown Request

```
<TeardownRequestData>  
<streamNID><!-- Unique identifier for the stream --></streamNID>  
</TeardownRequestData>
```

### Example 2.4. Teardown Response

```
<TeardownResponseData>  
<streamNID><!-- Unique identifier for the stream --></streamNID>  
</TeardownResponseData>
```

## Maintaining state for the Stream

The device has to maintain state for each of Stream, it relies on events from the colocated media engine and the peering source

- The state of the stream is updated based on

The presence of the MediaSource

Events from colocated media engine

The Start Response AgentMessages received from the MediaSource

The Events received from the MediaSource

- The state machine action handlers are responsible for

Restarting a stream once an absent MediaSource comes back online

Publishing StreamStatusUpdates to the SC whenever the state of the stream changes

### Example 2.5. StreamStatusEvent payload sent to the central server

```
<StreamStatusEventData>  
<StreamStatus .../>  
</StreamStatusEventData>
```

## Presence Events received by Devices

A Entity is a source or a destination These requests are typically received by an entity which has setup/needs to setup communications with a peer. This setup is done in the context of a "MediaRoom". The

"MediaRoom" has a unique identifier referred to as roomNID. All resource allocations on the source/dest entities can be tied/linked to the presence of the MediaRoom Please check the API-MediaService.pdf document for further details on the Presence Event