ALLSEEN ALLIANCE

AllJoyn™ Audio Service Framework 1.0 Interface Definition

June 30, 2014

Contents

Figures

Tables

1 Introduction

1.1 Purpose

This document specifies AllJoyn™ interfaces and objects to stream audio between devices.

1.2 Scope

This document is intended for developers of AllJoyn audio streaming applications who have some familiarity with audio programming.

1.3 Release history

Release version	What changed
Pre-14.06	The following interfaces were added: ■ Stream interface version 1 ■ Stream.Port interface version 1 ■ Stream.Port.AudioSource interface version 1 ■ Stream.Port.AudioSink interface version 1 ■ Stream.Port.ImageSource ■ Stream.Port.ImageSink ■ Stream.Port.Application.MetadataSource ■ Stream.Port.Application.MetadataSink ■ Stream.Clock
14.06	The following changes were made to this document: ■ Updated the document title and Overview chapter title (changed Specification to Definition). ■ Added a Mandatory column for method and signal parameters to support the AllSeen Alliance Compliance and Certification program. ■ Added a note in the Specification Overview chapter to address the AllSeen Alliance Compliance and Certification program. ■ Included the object path for each interface. ■ Added the Close method to the Stream interface. ■ Updated the following data points: ❒ Connect method's path parameter signature (o) ❒ Pause method's parameter name (timeNanos) ❒ Flush method's input parameter name and specified output parameter name (numBytesFlushed) ❒ Data signal's parameter name (bytes) for the AudioSource and ImageSource interfaces

Release version	What changed
	❒ Data signal's parameter name (dictionary) for the MetadataSource interface ❒ AdjustTime method parameter name (adjustNanos) ■ Separated the following interfaces to address the information unique to the relevant Sink and Source interfaces: ❒ Stream.Port.Audio ❒ Stream.Port.Image ❒ Stream.Port.ApplicationMetadata ■ Added the Control.Volume interface. ■ Updated the Introspection XML to reflect the Control.Volume output.

1.4 References

Except for supporting information, the following are reference documents found on the AllSeen Alliance web site's Docs/Downloads section.

- *AllJoyn™ Framework Tutorial*
- *Introduction to AllJoyn™ Thin Client*
- *Apple Lossless Audio Codec* (*http://alac.macosforge.org/*)
- *AllJoyn™ Data Type Signature*

Documentation supporting the About feature is listed on the Service Framework tab for the supported platform, such as Android, LInux, Thin Client, etc.

- AllJoyn™ About Feature 1.0 Interface Specification

1.5 Acronyms and terms

Term	Definition
Elementary stream	A data stream of one type of media (audio, image, or metadata)
FIFO	First In, First Out
PCM	Pulse Control Modulation

2 Definition Overview

A stream consists of one or more elementary streams; an elementary stream consists of one type of media (audio, image, or metadata).

Note All methods and signals are considered mandatory to support the AllSeen Alliance Compliance and Certification program. Individual parameters for a given method or signal may be considered mandatory or optional, and are specified accordingly in this document.

3 Stream Object

A stream object is a BusObject implementing the org.alljoyn.Stream interface. It has one child port object for each elementary stream. A port object is a BusObject implementing the org.alljoyn.Stream.Port interface and one of the media-type specific port interfaces (org.alljoyn.Stream.Port.AudioSink, etc.). Port objects send or receive elementary streams.

Table 1 lists example objects and interfaces used by a speaker. *Table 2* lists example objects and interfaces used by a media player.

- /Speaker/In is a stream object
- /Speaker/In/Audio, /Speaker/In/Image, and /Speaker/In/Metadata are child port objects.

Table 1: Example objects and interfaces implemented by a speaker

Object path	Interfaces implemented
/Speaker/In	org.alljoyn.Stream
/Speaker/In/Audio	■ org.alljoyn.Stream.Port ■ org.alljoyn.Stream.Port.AudioSink
/Speaker/In/Image	■ org.alljoyn.Stream.Port ■ org.alljoyn.Stream.Port.ImageSink
/Speaker/In/Metadata	■ org.alljoyn.Stream.Port ■ org.alljoyn.Stream.Port.Application.MetadataSink

Table 2: Example objects and interfaces implemented by a media player

Object path	Interfaces implemented
/Player/Out	org.alljoyn.Stream
/Player/Out/Audio	■ org.alljoyn.Stream.Port ■ org.alljoyn.Stream.Port.AudioSource
/Player/Out/Image	■ org.alljoyn.Stream.Port ■ org.alljoyn.Stream.Port.ImageSource
/Player/Out/Metadata	■ org.alljoyn.Stream.Port ■ org.alljoyn.Stream.Port.Application.MetadataSource

Figure 1 illustrates how source ports are connected to sink ports to enable streaming.

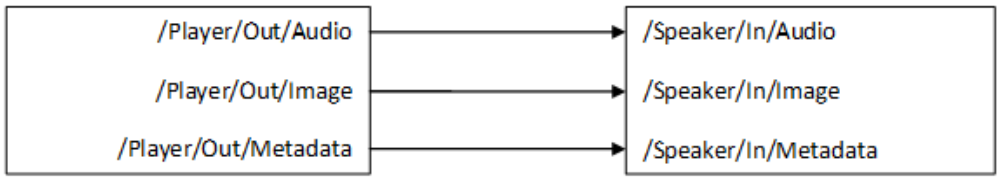

Figure 1: Connected media player and speaker

4 Typical Stream Flows

4.1 One source and one sink

Figure 2 illustrates a typical flow for one source and one sink.

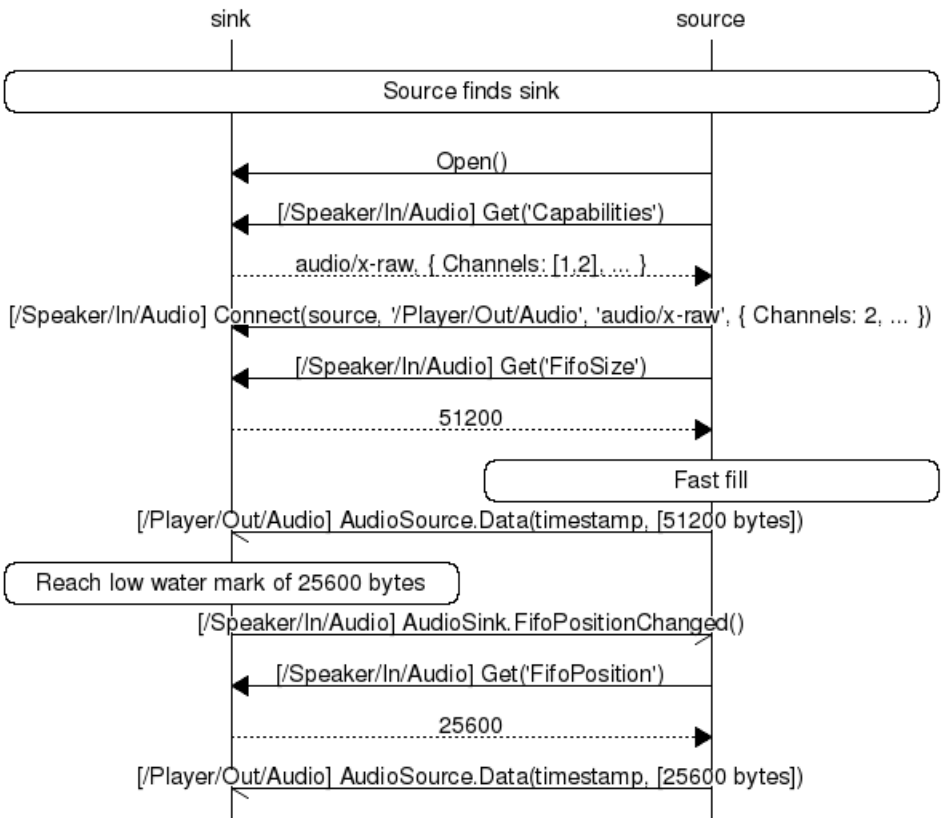

Figure 2: Typical call flow for one source and one sink

4.2 One source and two sinks

Figure 3 illustrates the typical flow for one source and two sinks.

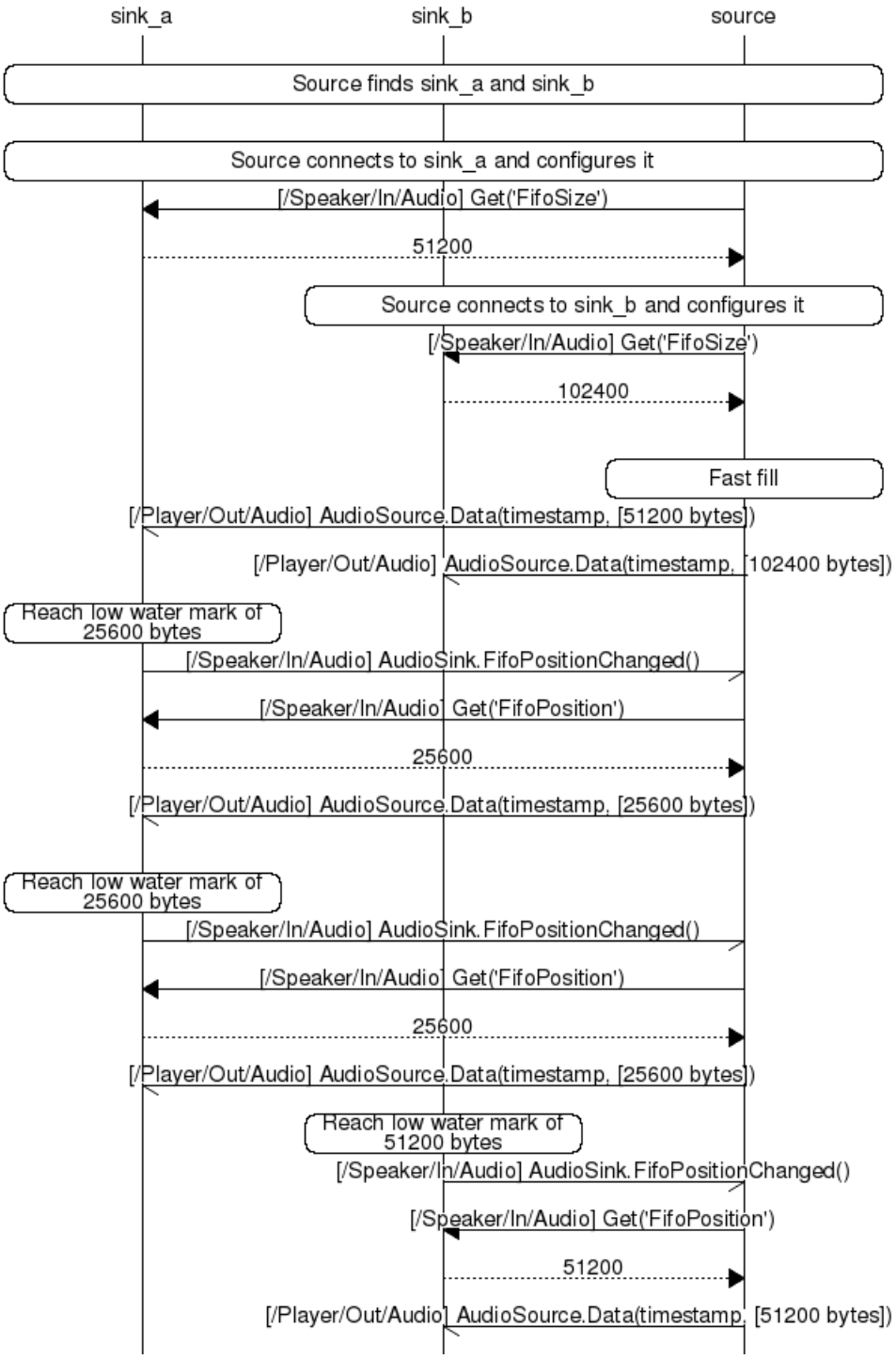

Figure 3: Typical call flow for one source and two sinks

5 Discovery

To be discovered by interested client applications on other devices, an audio implementation advertises its existence using the About feature. Refer to the *AllJoyn™ About Feature Interface Specification* document for more information.

5.1 BusObject paths

Implementations of audio should publish the object paths of the BusObjects that implement org.alljoyn.Stream and the object paths of BusObjects that implement org.alljoyn.Stream.Port interfaces using the org.alljoyn.About interface. See *Table 1* and *Table 2* for examples of published BusObjects.

5.2 Session port value

In addition to the object paths, the implementation should also publish the session port value that the service uses to listen for incoming client connections. Use this port value in the “port” parameter of the org.alljoyn.About.Announce signal.

6 Stream Interface

The Stream interface is responsible for stream creation and control over a stream's ports.

6.1 Interface name

Interface name	Version	Secured	Object path
org.alljoyn.Stream	1	no	Any object path

6.2 Properties

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

6.3 Methods

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

6.3.1 Open

Returns an error if not implemented or called more than once.

Inputs

None.

Output

None.

Description

Open the stream.

6.3.2 Close

Returns an error if not implemented or called more than once.

Inputs

None.

Outputs

None.

Description

Close the stream.

7 Stream.Port Interface

The Stream.Port interface is responsible for control over an elementary stream. A port object implements this interface together with a media-specific port interface such as Stream.Port.AudioSource or Stream.Port.AudioSink.

7.1 Interface name

Interface name	Version	Secured	Object path
org.alljoyn.Stream.Port	1	no	Child node of node implementing the Stream interface

7.2 Properties

Property name	Signature	List of values	Writable	Description
Version	q	Positive integers	no	Interface version number
Direction	y	■ 0 - sink ■ 1 - source	no	Indicates if this port is a source or sink port. Source ports send elementary streams, sink ports receive them.
Capabilities	a(sa{sv})	See *Media Types* for more information.	no	Defines the supported capabilities of this port. The capabilities of unknown media types should be ignored.

7.3 Methods

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

7.3.1 Connect

Inputs

Parameter name	Mandatory	Signature	List of values	Description
host	yes	s	AllJoyn name	The AllJoyn name of the remote stream port host to connect to.
path	yes	o	Object path	The AllJoyn object path of the remote port on the host to connect to.

Parameter name	Mandatory	Signature	List of values	Description
configuration	yes	(sa{sv})	See *Media Types* for more information.	A media type and the values to configure its parameters with.

Output

None.

Description

Connect this port to a remote port and configures the ports to send or receive an elementary stream.

If the configuration is not compatible with the capabilities, this method should return an error.

If this port is already connected to the remote port, this method should return an error. To reconfigure a connected port, first close the stream, then open and connect with the new configuration.

7.4 Signals

Signal name	Parameters			Session-less	Description
OwnershipLost	**Name**	**Mandatory**	**Signature**	no	The port emits this signal to the currently connected remote port when it connects to a new remote port. newOwner - AllJoyn name of the new remote port.
	newOwner	yes	s		

8 Stream.Port.Audio Interfaces

The Stream.Port.AudioSource and Stream.Port.AudioSink interfaces define the data format for audio/* media. The interfaces are implemented by port objects that support audio/* capabilities.

8.1 FIFO control

An audio sink exposes a FIFO to receive streamed data from an audio source. Control of the FIFO is implemented as a state machine. *Figure 4* illustrates the states of the FIFO.

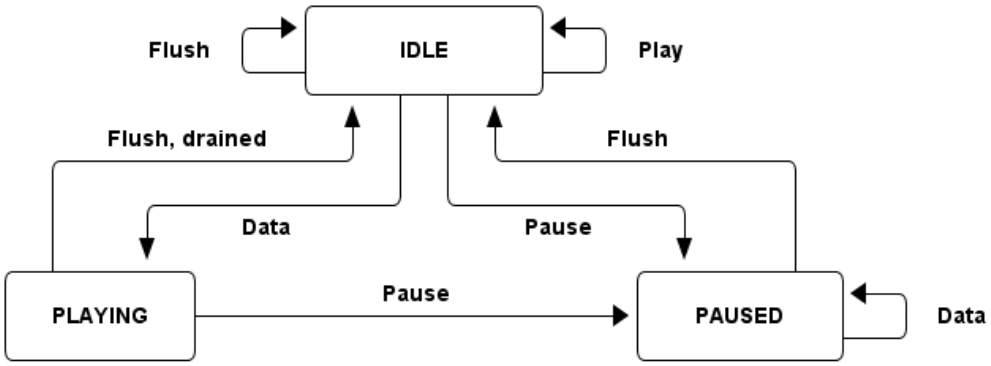

Figure 4: Audio sink state diagram

Filling an audio sink's FIFO is accomplished using the Data signal, FifoSize and FifoPosition properties, and FifoPositionChanged signal. On receipt of each FifoPositionChanged signal, the audio source can send (FifoSize – FifoPosition) more bytes of data. The audio source should not send more than this; doing so can lead to blocking the receipt of other BusMethods and BusSignals by the audio sink, or the audio sink discarding the data.

Use the Flush method to tell the audio sink to discard all the data in its FIFO in preparation for new data.

8.2 Synchronization

If an audio sink supports synchronized playback (by implementing the Stream.Clock interface), use the timestamp parameter of the Data signal to determine the correct time to render the data. The audio source computes a timestamp sufficient to ensure that it is greater than the network latency plus the rendering latency of each audio sink. Audio sinks should discard data that cannot be rendered in time.

8.3 Stream.Port.AudioSink interface

The following section details the Stream.Port.AudioSink interface.

8.3.1 Interface name

Interface name	Version	Secured	Object path
org.alljoyn.Stream.Port.AudioSink	1	no	Child node of node implementing the Stream interface

8.3.2 Properties

Property name	Signature	List of values	Writable	Description
Version	q	Positive integers	no	Interface version number
FifoSize	u	Size in bytes	no	The size of the sink's FIFO in bytes. The size in samples may be computed from the configured Channels, Format, and Rate.
FifoPosition	u	Position in bytes	no	The current position of the FIFO in bytes. Bytes read from this position are submitted to the audio device.
Delay	(uu)	■ Position in bytes ■ Size in bytes	no	Use the sum of these values and the configured Channels, Format, and Rate properties to compute the delay in seconds from when a sample is received to when it is heard. ■ The first value is FifoPosition. ■ The second value is the size of the audio device's FIFO in bytes.
PlayState	y	■ 0 - idle ■ 1 - playing ■ 2 - paused	no	The current rendering state.

8.3.3 Methods

The following methods provide control of the audio sink's FIFO state machine.

8.3.3.1 Flush

Inputs

Parameter name	Mandatory	Signature	List of values	Description
timeNanos	yes	t	Timestamp in nanoseconds since the UNIX epoch	Timestamp in nanoseconds since the UNIX epoch to flush at. ■ If an audio sink does not support synchronized playback (by not implementing the Stream.Clock interface), this value should be 0 and the sink should immediately flush. ■ If an audio sink supports synchronized playback, this method should not return a reply until the flush is complete.

Output

Parameter name	Mandatory	Return signature	Description
numBytesFlushed	yes	u	Number of bytes flushed from the FIFO. When timeNanos is not 0, this value is the number of bytes flushed after the flush is complete.

Description

Flush the FIFO of this instance. FifoPositionChanged should be emitted after the flush is complete.

8.3.3.2 Pause

Inputs

Parameter name	Mandatory	Signature	List of values	Description
timeNanos	yes	t	Timestamp in nanoseconds since the UNIX epoch	The timestamp in nanoseconds since the UNIX epoch to stop rendering data at. ■ If an audio sink does not support synchronized playback (by not implementing the Stream.Clock interface), this value should be 0 and the audio sink should immediately stop rendering. ■ If an audio sink supports synchronized playback, this method should not return a reply until the pause is complete.

Output

None.

Description

Tell the port to stop rendering data from the FIFO. The FIFO is not flushed.

8.3.3.3 Play

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

Inputs

None.

Output

None.

Description

Tell the port to start rendering data from the FIFO.

If the configuration is not compatible with the capabilities, this method should return an error.

8.3.4 Signals

Signal name	Parameters			Session-less	Description
FifoPositionChanged	None			no	An audio sink emits this signal when its FIFO position crosses the low watermark.
PlayStateChanged	**Name**	**Mandatory**	**Signature**	no	Emitted when the PlayState property changes. ■ oldState - Previous PlayState value. ■ newState - Current PlayState value.
	oldState	yes	y		
	newState	yes	y		

8.4 Stream.Port.AudioSource interface

The following section details the Stream.Port.AudioSource interface.

8.4.1 Interface name

Interface name	Version	Secured	Object path
org.alljoyn.Stream.Port.AudioSource	1	no	Child node of node implementing the Stream interface

8.4.2 Properties

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

8.4.3 Methods

None.

8.4.4 Signals

Signal name	Parameters			Sessionless	Description
Data	**Parameter name**	**Mandatory**	**Signature**	no	This signal is sent by the audio source to the audio sink. ■ timestamp - the timestamp in nanoseconds since the UNIX epoch to render the data. If an audio sink does not support synchronized playback (by not implementing the Stream.Clock interface), this value should be 0. ■ bytes - the interleaved sample data.
	timestamp	yes	t		
	bytes	yes	ay		

9 Stream.Port.Image Interfaces

The Stream.Port.ImageSource and Stream.Port.ImageSink interfaces define the data format for image/* media types. The interfaces are implemented by port objects that support image/* capabilities.

9.1 Stream.Port.ImageSink interface

The following section details the Stream.Port.ImageSink interface.

9.1.1 Interface name

Interface name	Version	Secured	Object path
org.alljoyn.Stream.Port.ImageSink	1	no	Child node of the node implementing the Stream interface

9.1.2 Properties

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

9.2 Stream.Port.ImageSource interface

The following section details the Stream.Port.ImageSource interface.

9.2.1 Interface name

Interface name	Version	Secured	Object path
org.alljoyn.Stream.Port.ImageSource	1	no	Child node of the node implementing the Stream interface

9.2.2 Properties

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

9.2.3 Signals

Signal name	Parameters			Session-less	Description
Data	**Name**	**Mandatory**	**Signature**	no	This signal is sent by the source to the sink. bytes - Segment of the image data.
	bytes	yes	ay		

10 Stream.Port.Application.Metadata Interfaces

The Application.MetadataSource and Application.MetadataSink interfaces define the data format for application/x-metadata media types. The interfaces are implemented by port objects that support the application/x-metadata capability.

10.1 Stream.Port.Application.MetadataSink interface

The following section details the Stream.Port.Application.MetadataSink interface.

10.1.1 Interface name

Interface name	Version	Secured	Object path
org.alljoyn.Stream.Application.MetadataSink	1	no	Child node of the node implementing the Stream interface

10.1.2 Properties

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

10.2 Stream.Port.Application.MetadataSource interface

The following section details the Stream.Port.Application.MetadataSource interface.

10.2.1 Interface name

Interface name	Version	Secured	Object path
org.alljoyn.Stream.Port. Application.MetadataSource	1	no	Child node of the node implementing the Stream interface

10.2.2 Properties

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

10.2.3 Signals

Signal name	Parameters			Session less	Description
Data	**Name**	**Mandatory**	**Signature**	no	This signal is sent by the source to the sink. dictionary - Metadata key/value pairs. See *Media Types* for metadata keys and relevant values.
	dictionary	yes	a{sv}		

11 Control.Volume Interface

11.1 Interface name

Interface name	Version	Secured	Object path
org.alljoyn.Control.Volume	1	no	Must be the same as the path implementing the AudioSink interface

11.2 Properties

Property name	Signature	List of values	Writable	Description
Version	q	Positive integers	no	Interface version number
Volume	n	Signed integers	yes	Current volume of this device
VolumeRange	(nnn)	Signed integers ■ high ■ low ■ step	no	Maximum (high) and minimum (low) values of the volume. The step value is the incremental unit, the value of Volume is always a multiple of the step.
Mute	b	■ true ■ false	yes	Whether or not this device is muted
Enabled	b	■ true ■ false	no	Whether or not Volume control is enabled. If this value is false, all methods that change the playback volume will return an error.

11.3 Methods

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

11.3.1 AdjustVolume

Inputs

Parameter name	Mandatory	Signature	List of values	Description
delta	yes	n	Signed integer	The amount by which to increase or decrease the volume.

Output

None.

Description

Adjust the volume by a certain amount. The new volume will be equal to old volume + delta.

11.3.2 AdjustVolumePercent

Input

Parameter name	Mandatory	Signature	List of values	Description
change	yes	d	double precision floating point value	For values greater than 0 and smaller than 1, the percentage by which to raise the volume. ■ For values smaller than 0 and bigger than -1, the percentage by which to reduce the volume. ■ For values greater than or equal to 1, increase volume to maximum. ■ For values smaller than or equal to -1, decrease volume to the minimum.

Output

None.

Description

Adjust the volume by a certain percentage.

11.4 Signals

Signal name	Parameters			Sessionless	Description
VolumeChanged	**Parameter name**	**Mandatory**	**Signature**	no	Signal sent when the volume has changed.

Signal name	Parameters			Sessionless	Description
	newVolume	yes	n		
MuteChanged	**Parameter name**	**Mandatory**	**Signature**	no	Signal sent when the volume's mute state changes
	newMute	yes	b		
EnableChanged	**Parameter name**	**Mandatory**	**Signature**	no	Signal sent when the volume control is enabled or disabled
	enabled	yes	b		

12 Stream.Clock Interface

The Stream.Clock interface provides the ability to set the clock used for synchronized playback. It can be implemented by a stream object.

The following sequence defines the clock synchronization process.

1. Record the current time as t0.
2. Call SetTime(t0) and wait for the method reply.
3. Record the current time as t1.
4. Call AdjustTime((t1 – t0) / 2).

The maximum error of the clock skew is the value passed to AdjustTime, (t1 – t0) / 2.

12.1 Interface name

Interface name	Version	Secured	Object path
org.alljoyn.Stream.Clock	1	no	Child node of the node implementing the Stream interface

12.2 Properties

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

12.3 Methods

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

12.3.1 SetTime

Inputs

Parameter name	Mandatory	Signature	List of values	Description
timeNanos	yes	t	Timestamp in nanoseconds since the UNIX epoch	The time to set the clock used for synchronized playback.

Output

None.

Description

Set the time of the clock used for synchronized playback.

12.3.2 AdjustTime

Inputs

Parameter name	Mandatory	Signature	List of values	Description
adjustNanos	yes	x	Positive or negative time in nanoseconds	The amount, positive or negative, to adjust the time.

Output

None.

Description

Adjusts the time of the clock used for synchronized playback forwards or backwards from its current value.

13 Media Types

The following sections define the values of the (sa{sv}) used in the Stream.Port.Capabilities and Stream.Port.Connect interfaces.

- For a capability, the first parameter of the struct is the media type (such as audio/x-raw). The second parameter is a dictionary of configurable parameters for the media type. The value of the dictionary entry for a configurable parameter is an array of the supported values of the configurable parameter.
- A configuration is identical to a capability except that the value of the dictionary entry for a configurable parameter is only one of the supported values listed in the capability.

In addition to the media types defined below, additional media types may be defined in future revisions of this specification or in addition to this specification.

13.1 audio/x-raw

Raw interleaved PCM data. If audio is supported, this media type is mandatory.

Key	Signature	List of values	Description
Channels	y	Positive integer	Number of channels. Support for 1 and 2 channels is mandatory.
Format	s	Sample format	Sample formats. s16le – PCM signed 16-bit wide little endian samples. Support for s16le is mandatory.
Rate	q	Sample rate	Sample rate. Support for 44100 and 48000 sample rates is mandatory.

13.2 audio/x-alac

Apple Lossless. Support for this media type is optional.

Key	Signature	List of values	Description
Channels	y	Positive integer	Number of channels. Support for 1 and 2 channels is mandatory.
Format	s	Sample format	Sample formats. s16le – PCM signed 16-bit wide little endian samples. Support for s16le is mandatory.
Rate	q	Sample rate	Sample rate. Support for 44100 and 48000 sample rates is mandatory.
MagicCookie	ay	Array of bytes	Magic cookie. Used only in configuration.

Key	Signature	List of values	Description
FramesPerPacket	u	Frames per packet	Frames per packet. Used only in configuration.

13.3 image/jpeg

JPEG image data. Support for this media type is optional.

13.4 application/x-metadata

Stream metadata. Support for this media type is optional.

14 Media Item Keys

Table 3 lists the keys and values that can be used in a metadata dictionary entry.

Date values are to only be represented as strings in one of the following ISO 8601 formats:

- Year only: “<year>” where <year> includes all digits of the year.
- Date: “<year><month><day>” where:
  - <year> includes all digits of the year
  - <month> is a 2-digit representation of month (“01” = January)
  - <day> is the day of the month (e.g., “09”).
- Date and time: “<year><month><day>T<hour><minute><second>±<offset>” where:
  - <year>, <month>, and <day> are the same as the date previously specified
  - <hour> is the hour (“00” through “23”)
  - <minute> is the minute (“00” through “59”)
  - <second> is the second (“00” through “59”)
  - <offset> is the offset from UTC (“-0800” corresponds to Pacific Standard Time)
- Date and time: “<year><month><day>T<hour><minute>±<offset>” is the same as the previous date and time specification except without the <second> (seconds) portion.

Table 3: Media metadata keys and values

Key	Signature	List of Values	Description
Name	s		Name or title of the item
Album	s		Album title
AlbumArtist	s		Album artist
AlbumRating	Y	0-5	“Star” rating
Artist	s		Artist that performed the item
Bpm	g		Beats per minute
Compilation	b		Indicates if item is part of a compilation
Composer	s		Composer of the item
DiscCount	g		Total number of discs in a multi-disc release
DiscNumber	g		Disc number in a multi-disc release
Duration	u		Duration of the item in milliseconds
Genre	s		Genre of the item
Rating	y	0-5	“Star” rating
ReleaseDate	s	Date	Date item was first released

Key	Signature	List of Values	Description
TrackCount	g		Total number of tracks on the album
TrackNumber	g		Track number of the item within the album

15 AllJoyn Introspection XML

The following XML defines the audio streaming interfaces.

```
<node xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.allseenalliance.org/schemas/introspect.xsd">

<interface name="org.alljoyn.Stream">
  <property name="Version" type="q" access="read"/>
  <method name="Open"/>
  <method name="Close"/>
</interface>

<interface name="org.alljoyn.Stream.Port">
  <property name="Version" type="q" access="read"/>
  <property name="Direction" type="y" access="read"/>
  <property name="Capabilities" type="a(sa{sv})" access="read"/>
  <signal name="OwnershipLost">
    <arg name="newOwner" type="s"/>
  </signal>
  <method name="Connect">
    <arg name="host" type="s" direction="in"/>
    <arg name="path" type="o" direction="in"/>
    <arg name="configuration" type="(sa{sv})" direction="in"/>
  </method>
</interface>

<interface name="org.alljoyn.Stream.Port.AudioSink">
  <property name="Version" type="q" access="read"/>
  <property name="FifoSize" type="u" access="read"/>
  <property name="FifoPosition" type="u" access="read"/>
  <property name="Delay" type="(uu)" access="read"/>
  <signal name="FifoPositionChanged"/>
  <property name="PlayState" type="y" access="read"/>
  <signal name="PlayStateChanged">
    <arg name="oldState" type="y"/>
    <arg name="newState" type="y"/>
  </signal>
  <method name="Play"/>
  <method name="Pause">
    <arg name="timeNanos" type="t" direction="in"/>
  </method>
  <method name="Flush">
    <arg name="timeNanos" type="t" direction="in"/>
    <arg name="numBytesFlushed" type="u" direction="out"/>
  </method>
</interface>

<interface name="org.alljoyn.Stream.Port.AudioSource">
```

```
  <property name="Version" type="q" access="read"/>
  <signal name="Data">
    <arg name="timestamp" type="t"/>
    <arg name="bytes" type="ay"/>
  </signal>
</interface>

<interface name="org.alljoyn.Stream.Port.ImageSink">
  <property name="Version" type="q" access="read"/>
</interface>

<interface name="org.alljoyn.Stream.Port.ImageSource">
  <property name="Version" type="q" access="read"/>
  <signal name="Data">
    <arg name="bytes" type="ay"/>
  </signal>
</interface>

<interface name="org.alljoyn.Stream.Port.Application.MetadataSink">
  <property name="Version" type="q" access="read"/>
</interface>

<interface name="org.alljoyn.Stream.Port.Application.MetadataSource">
  <property name="Version" type="q" access="read"/>
  <signal name="Data">
    <arg name="dictionary" type="a{sv}"/>
  </signal>
</interface>

<interface name="org.alljoyn.Stream.Clock">
  <property name="Version" type="q" access="read"/>
  <method name="SetTime">
    <arg name="timeNanos" type="t" direction="in"/>
  </method>
  <method name="AdjustTime">
    <arg name="adjustNanos" type="x" direction="in"/>
  </method>
</interface>

<interface name="org.alljoyn.Control.Volume">
  <property name="Version" type="q" access="read"/>
  <property name="Volume" type="n" access="readwrite"/>
  <property name="VolumeRange" type="(nnn)" access="read"/>
  <property name="Mute" type="b" access="readwrite"/>
  <signal name="VolumeChanged">
    <arg name="newVolume" type="n"/>
  <signal name="MutedChanged">
    <arg name="newMute" type="b"/>
  </signal>
  <method name="AdjustVolume">
    <arg name="delta" type="n" direction="in"/>
```

```
  </method>
  <method name="AdjustVolumePercent">
    <arg name="change" type="d" direction="in"/>
  </method>
  <property name="Enabled" type="b" access="read"/>
  <signal name="EnableChanged">
    <arg name="enabled" type="b"/>
  </signal>
</interface>


</node>
```