AllJoyn™ Audio Service Framework 1.0 Test Case Specifications

May 1, 2014

Contents

1 Introduction 3

1.1 Purpose 3
1.2 Scope 3
1.3 References 3

2 Environment setup 4

2.1 Requirements 4
2.2 Preconditions 4
2.3 Parameters 4

3 Audio service framework test cases 5

3.1 Audio-v1-01: Validate Stream bus objects 5
3.2 Audio-v1-02: Opening a Stream bus object 5
3.3 Audio-v1-03: Opening and closing a Stream bus object 6
3.4 Audio-v1-04: Closing an unopened Stream bus object 6
3.5 Audio-v1-05: Verify any AudioSink capabilities 7
3.6 Audio-v1-06: Verify any ImageSink capabilities 8
3.7 Audio-v1-07: Verify any Application.MetadataSink capabilities 9
3.8 Audio-v1-08: Configure any AudioSink port 9
3.9 Audio-v1-09: Configure any AudioSink bus object with an invalid configuration 10
3.10 Audio-v1-10: Configure any AudioSink bus object twice 11
3.11 Audio-v1-11: Check for OwnershipLost signal 12
3.12 Audio-v1-12: Playback on an AudioSink 13
3.13 Audio-v1-13: Pausing playback on an AudioSink 14
3.14 Audio-v1-14: Flushing a paused AudioSink 15
3.15 Audio-v1-15: Flushing a playing AudioSink 17
3.16 Audio-v1-16: Paused AudioSink remains paused after sending data 18
3.17 Audio-v1-17: Playing an empty AudioSink remains IDLE 19
3.18 Audio-v1-18: Flushing an idle AudioSink 20
3.19 Audio-v1-19: Sending data to an ImageSink 21
3.20 Audio-v1-20: Sending data to an Application.MetadataSink 22
3.21 Audio-v1-21: Setting the mute state on an AudioSink 23
3.22 Audio-v1-22: Setting the volume on an AudioSink 25
3.23 Audio-v1-23: Setting an invalid volume on an AudioSink 27
3.24 Audio-v1-24: Independence of mute and volume on an AudioSink 28
3.25 Audio-v1-25: Synchronize clocks on an AudioSink 30

1 Introduction

1.1 Purpose

These test cases evaluate and verify the functionality related to an AllJoyn™ Audio service framework AudioSink implementation exposed by a device through the Audio 1.0 interfaces.

The Audio service framework allows for the streaming of audio and associated images/metadata between a source and one or more sinks.

1.2 Scope

These test cases are designed to determine if a device conforms to the Audio interface specifications. Successful completion of all test cases does not guarantee that the tested device will interoperate with other devices.

These test cases must be run for each bus object implementing the org.alljoyn.Stream interface on a DUT as advertised in the About announcements received from the DUT.

1.3 References

The following are reference documents.

- *AllJoyn™ Audio Service Framework 1.0 Interface Specification*

2 Environment setup

2.1 Requirements

The following are required to execute these test cases:

- An AllJoyn device (the device under test or DUT) which implements an AudioSink according to the Audio service framework 1.0 specification.
- 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
- The DUT is announcing its capabilities including support for the Stream interface through its About announcement.

2.3 Parameters

Table 1. Parameters for the Audio service framework

Parameter	Description
DeviceId	Device ID of the DUT
AppId	Application ID of the application on the DUT
TestObjectPath	The path of the Stream bus object being tested

3 Audio service framework test cases

3.1 Audio-v1-01: Validate Stream bus objects

Objective

Verify that the application on the DUT exposes the proper interfaces and versions required of an AudioSink device.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About Announcement from the application, the test device joins a session with the application at the port specified in the received About announcement.
3. The test device retrieves the Version property of the Stream bus object specified by the TestObjectPath test parameter.
4. The test device uses the Introspect() method on the Introspectable interface of the Stream bus object to retrieve the list of the child bus objects and their interfaces.
5. The test device retrieves the Version and Direction properties for each of the child bus objects implementing the Port interface.
6. The test device leaves the session.

Expected results

- The bus object implementing the Stream interface has a Version property of 1.
- The bus object implementing the Stream interface has at least one child bus object implementing org.alljoyn.Stream.Port.
- Each child bus object implementing the Port interface has a Version property of 1 and a Direction property of 0.
- Each child bus object implementing the Port interface also implements one of the media type-specific Port interfaces:
  - org.alljoyn.Stream.Port.AudioSink
  - org.alljoyn.Stream.Port.ImageSink
  - org.alljoyn.Stream.Port.Application.MetadataSink

3.2 Audio-v1-02: Opening a Stream bus object

Objective

Verify that the test device can open the Stream bus object.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About Announcement from the application, the test device joins a session with the application at the port specified in the received About announcement.
3. The test device calls the Open() method on the Stream bus object specified by the TestObjectPath test parameter.
4. The test device leaves the session.

Expected results

The Open() method call returns successfully without an error.

3.3 Audio-v1-03: Opening and closing a Stream bus object

Objective

Verify that the test device can open and close the Stream bus object.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About Announcement from the application, the test device joins a session with the application at the port specified in the received About announcement.
3. The test device calls the Open() method on the Stream bus object specified by the TestObjectPath test parameter.
4. The test device calls the Close() method on the Stream bus object.
5. The test device leaves the session.

Expected results

- The Open() method call returns successfully without an error.
- The Close() method call returns successfully without an error.

3.4 Audio-v1-04: Closing an unopened Stream bus object

Objective

Verify that an error is received when closing an unopened Stream bus object.

Procedure

1. The test device listens for an About announcement from the application on the DUT.

2. After receiving an About Announcement from the application, the test device joins a session with the application at the port specified in the received About announcement.
3. The test device calls the Open() method on the Stream bus object specified by the TestObjectPath test parameter.
4. The test device calls the Close() method on the Stream bus object.
5. The test device calls the Close() method on the Stream bus object.
6. The test device leaves the session.

Expected results

- The Open() method call returns successfully without an error.
- The Close() method call returns successfully without an error.
- The second Close() method call returns an error.

3.5 Audio-v1-05: Verify any AudioSink capabilities

Objective

Verify that any AudioSink bus object supports the mandatory capabilities and each media type matches "audio/*".

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About Announcement from the application, the test device joins a session with the application at the port specified in the received About announcement.
3. The test device uses the Introspect() method on the Introspectable interface to locate an AudioSink child bus object of the Stream bus object specified by the TestObjectPath test parameter.
   - If one is not found, execution of the test case ends.
   - If one is found, the test device calls the Open() method on the Stream bus object.
4. The test device retrieves the Capabilities property on the AudioSink bus object.
5. The test device leaves the session.

Expected results

- If an AudioSink child bus object is not found, a note is added that the Stream does not support AudioSink.
- The Open() method call returns successfully without an error.
- The Capabilities property contains a capability with a mediaType of "audio/x-raw" and, at minimum, the following values are present in the array of supported values for each of the following configurable parameters:

  - Channels: 1, 2
  - Format: s16le
  - Rate: 44100, 48000
- If the Capabilities property contains a capability with a mediaType of "audio/x-alac", it has, at minimum, the following values are present in the array of supported values for each of the following configurable parameters:
  - Channels: 1, 2
  - Format: s16le
  - Rate: 44100, 48000
- For any capability in the Capabilities property, it has a mediaType matching "audio/*". If it provides values for any of these configurable parameters, the values are an array of values with signatures that match the following:
  - Channels: y
  - Format: s
  - Rate: q

3.6 Audio-v1-06: Verify any ImageSink capabilities

Objective

Verify that any ImageSink bus object contains at least one capability and that the media type of each capability matches "image/*".

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About Announcement from the application, the test device joins a session with the application at the port specified in the received About announcement.
3. The test device uses the Introspect() method on the Introspectable interface to locate an ImageSink child bus object of the Stream bus object specified by the TestObjectPath test parameter.
   - If one is not found, execution of the test case ends.
   - If one is found, the test device calls the Open() method on the Stream bus object.
4. The test device retrieves the Capabilities property on the ImageSink bus object.
5. The test device leaves the session.

Expected results

- If an ImageSink child bus object is not found, a note is added that the Stream does not support ImageSink.
- The Open() method call returns successfully without an error.

- The Capabilities property contains only capabilities with a mediaType matching "image/*".

3.7 Audio-v1-07: Verify any Application.MetadataSink capabilities

Objective

Verify that any Application.MetadataSink bus object contains one capability with a media type of "application/x-metadata".

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About Announcement from the application, the test device joins a session with the application at the port specified in the received About announcement.
3. The test device uses the Introspect() method on the Introspectable interface to locate an Application.MetadataSink child bus object of the Stream bus object specified by the TestObjectPath test parameter.
   - If one is not found, execution of the test case ends.
   - If one is found, the test device calls the Open() method on the Stream bus object.
4. The test device retrieves the Capabilities property on the Application.MetadataSink bus object.
5. The test device leaves the session.

Expected results

- If an Application.MetadataSink child bus object is not found, a note is added that the Stream does not support Application.MetadataSink.
- The Open() method call returns successfully without an error.
- The Capabilities property contains only one capability and that capability has a mediaType of "application/x-metadata".

3.8 Audio-v1-08: Configure any AudioSink port

Objective

Verify that any AudioSink can be configured with one of the mandatory configurations.

Procedure

1. The test device listens for an About announcement from the application on the DUT.

2. After receiving an About Announcement from the application, the test device joins a session with the application at the port specified in the received About announcement.
3. The test device uses the Introspect() method on the Introspectable interface to locate an AudioSink child bus object of the Stream bus object specified by the TestObjectPath test parameter.
   - If one is not found, execution of the test case ends.
   - If one is found, the test device calls the Open() method on the Stream bus object.
4. The test device calls the Connect() method on the AudioSink bus object with the following values for the parameters:
   - host: unique name of the test device's bus attachment
   - objectPath: "/Player/Out/Audio"
   - configuration: media type of "audio/x-raw" with the following parameters:
     - Channels: 1
     - Format: s16le
     - Rate: 44100
5. The test device calls the Close() method on the Stream bus object.
6. The test device leaves the session.

Expected results

- If an AudioSink child bus object is not found, a note is added that the Stream does not support AudioSink.
- The Open(), Connect(), and Close() method calls return successfully without an error.

3.9 Audio-v1-09: Configure any AudioSink bus object with an invalid configuration

Objective

Verify that any AudioSink bus object will return an error if configured with an invalid configuration.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About Announcement from the application, the test device joins a session with the application at the port specified in the received About announcement.

3. The test device uses the Introspect() method on the Introspectable interface to locate an AudioSink child bus object of the Stream bus object specified by the TestObjectPath test parameter.
   - If one is not found, execution of the test case ends.
   - If one is found, the test device calls the Open() method on the Stream bus object.
4. The test device calls the Connect() method on the AudioSink bus object with the following values for the parameters:
   - host: unique name of the test device's bus attachment
   - objectPath: "/Player/Out/Audio"
   - configuration: media type of "audio/x-unknown" with no parameters.
5. The test device calls the Close() method on the Stream bus object.
6. The test device leaves the session.

Expected results

- If an AudioSink child bus object is not found, a note is added that the Stream does not support AudioSink.
- The Open() and Close() method calls return successfully without an error.
- The Connect() method returns an error.

3.10 Audio-v1-10: Configure any AudioSink bus object twice

Objective

Verify that any AudioSink bus object will return an error if configured twice.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About Announcement from the application, the test device joins a session with the application at the port specified in the received About announcement.
3. The test device uses the Introspect() method on the Introspectable interface to locate an AudioSink child bus object of the Stream bus object specified by the TestObjectPath test parameter.
   - If one is not found, execution of the test case ends.
   - If one is found, the test device calls the Open() method on the Stream bus object.
4. The test device calls the Connect() method on the AudioSink bus object with the following values for the parameters:
   - host: unique name of the test device's bus attachment

- objectPath: “/Player/Out/Audio”
- configuration: media type of "audio/x-raw" with the following parameters:
  - Channels: 1
  - Format: s16le
  - Rate: 44100

5. The test device calls the Connect() method on the AudioSink bus object again with the same parameters.
6. The test device calls the Close() method on the Stream bus object.
7. The test device leaves the session.

Expected results

- If an AudioSink child bus object is not found, a note is added that the Stream does not support AudioSink.
- The Open() method calls return successfully without an error.
- The first Connect() method call returns successfully.
- The second Connect() method call returns an error.
- The Close() method calls return successfully without an error.

3.11 Audio-v1-11: Check for OwnershipLost signal

Objective

Verify that a change in ownership of the stream will result in an OwnershipLost signal.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About Announcement from the application, the test device joins a session with the application at the port specified in the received About announcement.
3. The test device uses the Introspect() method on the Introspectable interface to locate an AudioSink child bus object of the Stream bus object specified by the TestObjectPath test parameter.
   - If one is not found, execution of the test case ends.
   - If one is found, the test device calls the Open() method on the Stream bus object.
4. Using a secondary bus attachment, the test device joins another session with the application on the DUT, calls the Open() method and then the Close() method on the Stream bus object, and finally leaves that session.
5. The test device leaves the session.

Expected results

- If an AudioSink child bus object is not found, a note is added that the Stream does not support AudioSink.
- The Open() method calls return successfully without an error.
- Following the second sessions Open() method call, the first session receives an OwnershipLost signal with the unique name of the secondary bus attachment.
- The Close() method calls return successfully without an error.

3.12 Audio-v1-12: Playback on an AudioSink

Objective

Verify that sending data to an open and properly configured AudioSink will cause the audio data to be played.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About Announcement from the application, the test device joins a session with the application at the port specified in the received About announcement.
3. The test device uses the Introspect() method on the Introspectable interface to locate an AudioSink child bus object of the Stream bus object specified by the TestObjectPath test parameter.
   - If one is not found, execution of the test case ends.
   - If one is found, the test device calls the Open() method on the Stream bus object.
4. The test device calls the Connect() method on the AudioSink bus object with the following values for the parameters:
   - host: unique name of the test device’s bus attachment
   - objectPath: “/Player/Out/Audio”
   - configuration: media type of "audio/x-raw" with the following parameters:
     - Channels: 1
     - Format: s16le
     - Rate: 44100
5. The test device retrieves the FifoSize and FifoPosition properties on the AudioSink bus object.
6. The test device sends Data signals to the AudioSink bus object to fill the FIFO.
7. The test device waits to receive a PlayStateChanged signal from the AudioSink bus object.

8. The test device waits to receive a FifoPositionChanged signal from the AudioSink bus object.
9. The test device waits to receive a second PlayStateChanged signal from the AudioSink bus object.
10. The test device calls the Close() method on the Stream bus object.
11. The test device leaves the session.

Expected results

- If an AudioSink child bus object is not found, a note is added that the Stream does not support AudioSink.
- The Open() method calls return successfully without an error.
- The Connect() method calls return successfully without an error.
- After emitting the Data signal, the test device receives a PlayStateChanged signal with an oldState of 0 (IDLE) and a newState of 1 (PLAYING).
- The test device receives a FifoPositionChanged signal.
- The test device receives a PlayStateChanged signal with an oldState of 1 (PLAYING) and a newState of 0 (IDLE).
- The Close() method calls return successfully without an error.

3.13 Audio-v1-13: Pausing playback on an AudioSink

Objective

Verify that sending data to an open and properly configured AudioSink will cause the audio data to be played and that it can subsequently be paused.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About Announcement from the application, the test device joins a session with the application at the port specified in the received About announcement.
3. The test device uses the Introspect() method on the Introspectable interface to locate an AudioSink child bus object of the Stream bus object specified by the TestObjectPath test parameter.
   - If one is not found, execution of the test case ends.
   - If one is found, the test device calls the Open() method on the Stream bus object.
4. The test device calls the Connect() method on the AudioSink bus object with the following values for the parameters:
   - host: unique name of the test device’s bus attachment
   - objectPath: “/Player/Out/Audio”

- configuration: media type of "audio/x-raw" with the following parameters:
  - Channels: 1
  - Format: s16le
  - Rate: 44100

5. The test device retrieves the FifoSize and FifoPosition properties on the AudioSink bus object.
6. The test device sends Data signals to the AudioSink bus object to fill the FIFO.
7. The test device waits to receive a PlayStateChanged signal from the AudioSink bus object.
8. The test device calls the Pause() method on the AudioSink bus object with a timestamp of 0.
9. The test device waits to receive a PlayStateChanged signal from the AudioSink bus object.
10. The test device calls the Close() method on the Stream bus object.
11. The test device leaves the session.

Expected results

- If an AudioSink child bus object is not found, a note is added that the Stream does not support AudioSink.
- The Open() method calls return successfully without an error.
- The Connect() method calls return successfully without an error.
- After emitting the Data signal, the test device receives a PlayStateChanged signal with an oldState of 0 (IDLE) and a newState of 1 (PLAYING).
- After calling Pause(), the test device receives a PlayStateChanged signal with an oldState of 1 (PLAYING) and a newState of 2 (PAUSED).
- The Close() method calls return successfully without an error.

3.14 Audio-v1-14: Flushing a paused AudioSink

Objective

Verify that flushing a paused AudioSink will cause the FIFO to empty and the PlayState to change to IDLE.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About Announcement from the application, the test device joins a session with the application at the port specified in the received About announcement.

3. The test device uses the Introspect() method on the Introspectable interface to locate an AudioSink child bus object of the Stream bus object specified by the TestObjectPath test parameter.
   - If one is not found, execution of the test case ends.
   - If one is found, the test device calls the Open() method on the Stream bus object.
4. The test device calls the Connect() method on the AudioSink bus object with the following values for the parameters:
   - host: unique name of the test device's bus attachment
   - objectPath: "/Player/Out/Audio"
   - configuration: media type of "audio/x-raw" with the following parameters:
     - Channels: 1
     - Format: s16le
     - Rate: 44100
5. The test device retrieves the FifoSize and FifoPosition properties on the AudioSink bus object.
6. The test device sends Data signals to the AudioSink bus object to fill the FIFO.
7. The test device waits to receive a PlayStateChanged signal from the AudioSink bus object.
8. The test device calls the Pause() method on the AudioSink bus object with a timestamp of 0.
9. The test device waits to receive a PlayStateChanged signal from the AudioSink bus object.
10. The test device calls the Flush() method on the AudioSink bus object with a timestamp of 0.
11. The test device waits to receive a FifoPositionChanged signal from the AudioSink bus object.
12. The test device waits to receive a PlayStateChanged signal from the AudioSink bus object.
13. The test device calls the Close() method on the Stream bus object.
14. The test device leaves the session.

Expected results

- If an AudioSink child bus object is not found, a note is added that the Stream does not support AudioSink.
- The Open() method calls return successfully without an error.
- The Connect() method calls return successfully without an error.
- After emitting the Data signal, the test device receives a PlayStateChanged signal with an oldState of 0 (IDLE) and a newState of 1 (PLAYING).

- After calling Pause(), the test device receives a PlayStateChanged signal with an oldState of 1 (PLAYING) and a newState of 2 (PAUSED).
- After calling Flush(), the test device receives a PlayStateChanged signal with an oldState of 2 (PAUSED) and a newState of 0 (IDLE).
- The test device receives a FifoPositionChanged signal.
- The Close() method calls return successfully without an error.

3.15 Audio-v1-15: Flushing a playing AudioSink

Objective

Verify that flushing a playing AudioSink will cause the FIFO to empty and the PlayState to change to IDLE.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About Announcement from the application, the test device joins a session with the application at the port specified in the received About announcement.
3. The test device uses the Introspect() method on the Introspectable interface to locate an AudioSink child bus object of the Stream bus object specified by the TestObjectPath test parameter.
   - If one is not found, execution of the test case ends.
   - If one is found, the test device calls the Open() method on the Stream bus object.
4. The test device calls the Connect() method on the AudioSink bus object with the following values for the parameters:
   - host: unique name of the test device’s bus attachment
   - objectPath: “/Player/Out/Audio”
   - configuration: media type of "audio/x-raw" with the following parameters:
     - Channels: 1
     - Format: s16le
     - Rate: 44100
5. The test device retrieves the FifoSize and FifoPosition properties on the AudioSink bus object.
6. The test device sends Data signals to the AudioSink bus object to fill the FIFO.
7. The test device waits to receive a PlayStateChanged signal from the AudioSink bus object.
8. The test device calls the Flush() method on the AudioSink bus object with a timestamp of 0.

9. The test device waits to receive a FifoPositionChanged signal from the AudioSink bus object.
10. The test device waits to receive a PlayStateChanged signal from the AudioSink bus object.
11. The test device calls the Close() method on the Stream bus object.
12. The test device leaves the session.

Expected results

- If an AudioSink child bus object is not found, a note is added that the Stream does not support AudioSink.
- The Open() method calls return successfully without an error.
- The Connect() method calls return successfully without an error.
- After emitting the Data signal, the test device receives a PlayStateChanged signal with an oldState of 0 (IDLE) and a newState of 1 (PLAYING).
- After calling Flush(), the test device receives a PlayStateChanged signal with an oldState of 1 (PLAYING) and a newState of 0 (IDLE).
- The test device receives a FifoPositionChanged signal.
- The Close() method calls return successfully without an error.

3.16 Audio-v1-16: Paused AudioSink remains paused after sending data

Objective

Verify that an AudioSink will remain in the PAUSED state after more data is sent to it.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About Announcement from the application, the test device joins a session with the application at the port specified in the received About announcement.
3. The test device uses the Introspect() method on the Introspectable interface to locate an AudioSink child bus object of the Stream bus object specified by the TestObjectPath test parameter.
   - If one is not found, execution of the test case ends.
   - If one is found, the test device calls the Open() method on the Stream bus object.
4. The test device calls the Connect() method on the AudioSink bus object with the following values for the parameters:
   - host: unique name of the test device’s bus attachment
   - objectPath: “/Player/Out/Audio”

- configuration: media type of "audio/x-raw" with the following parameters:
  - Channels: 1
  - Format: s16le
  - Rate: 44100

5. The test device retrieves the FifoSize and FifoPosition properties on the AudioSink bus object.
6. The test device calls the Pause() method on the AudioSink bus object with a timestamp of 0.
7. The test device waits to receive a PlayStateChanged signal from the AudioSink bus object.
8. The test device sends Data signals to the AudioSink bus object to fill the FIFO.
9. The test device waits for a second to see if a PlayStateChanged signal is received from the AudioSink bus object.
10. The test device calls the Close() method on the Stream bus object.
11. The test device leaves the session.

Expected results

- If an AudioSink child bus object is not found, a note is added that the Stream does not support AudioSink.
- The Open() method calls return successfully without an error.
- The Connect() method calls return successfully without an error.
- After calling Pause(), the test device receives a PlayStateChanged signal with an oldState of 0 (IDLE) and a newState of 2 (PAUSED).
- After emitting the Data signal, the test device does not receive a PlayStateChanged signal.
- The Close() method calls return successfully without an error.

3.17 Audio-v1-17: Playing an empty AudioSink remains IDLE

Objective

Verify that an AudioSink will remain in the IDLE state after calling the Play() method if the FIFO contains no data.

Procedure

1. The test device listens for an About announcement from the application on the DUT.

2. After receiving an About Announcement from the application, the test device joins a session with the application at the port specified in the received About announcement.
3. The test device uses the Introspect() method on the Introspectable interface to locate an AudioSink child bus object of the Stream bus object specified by the TestObjectPath test parameter.
   - If one is not found, execution of the test case ends.
   - If one is found, the test device calls the Open() method on the Stream bus object.
4. The test device calls the Connect() method on the AudioSink bus object with the following values for the parameters:
   - host: unique name of the test device's bus attachment
   - objectPath: "/Player/Out/Audio"
   - configuration: media type of "audio/x-raw" with the following parameters:
     - Channels: 1
     - Format: s16le
     - Rate: 44100
5. The test device calls the Play() method on the AudioSink bus object.
6. The test device waits for a second to see if a PlayStateChanged signal is received.
7. The test device calls the Close() method on the Stream bus object.
8. The test device leaves the session.

Expected results

- If an AudioSink child bus object is not found, a note is added that the Stream does not support AudioSink.
- The Open() method calls return successfully without an error.
- The Connect() method calls return successfully without an error.
- After calling Play(), the test device does not receive a PlayStateChanged signal.
- The Close() method calls return successfully without an error.

3.18 Audio-v1-18: Flushing an idle AudioSink

Objective

Verify that calling Flush() on an AudioSink that is IDLE will result in a FifoPositionChanged signal.

Procedure

1. The test device listens for an About announcement from the application on the DUT.

2. After receiving an About Announcement from the application, the test device joins a session with the application at the port specified in the received About announcement.
3. The test device uses the Introspect() method on the Introspectable interface to locate an AudioSink child bus object of the Stream bus object specified by the TestObjectPath test parameter.
   - If one is not found, execution of the test case ends.
   - If one is found, the test device calls the Open() method on the Stream bus object.
4. The test device calls the Connect() method on the AudioSink bus object with the following values for the parameters:
   - host: unique name of the test device's bus attachment
   - objectPath: "/Player/Out/Audio"
   - configuration: media type of "audio/x-raw" with the following parameters:
     - Channels: 1
     - Format: s16le
     - Rate: 44100
5. The test device calls the Flush() method on the AudioSink bus object with a timestamp of 0.
6. The test device waits for a FifoPositionChanged signal from the AudioSink bus object.
7. The test device calls the Close() method on the Stream bus object.
8. The test device leaves the session.

Expected results

- If an AudioSink child bus object is not found, a note is added that the Stream does not support AudioSink.
- The Open() method calls return successfully without an error.
- The Connect() method calls return successfully without an error.
- After calling Flush(), the test device receives a FifoPositionChanged signal.
- The Close() method calls return successfully without an error.

3.19 Audio-v1-19: Sending data to an ImageSink

Objective

Verify that a data signal can be emitted to an ImageSink.

Procedure

1. The test device listens for an About announcement from the application on the DUT.

2. After receiving an About Announcement from the application, the test device joins a session with the application at the port specified in the received About announcement.
3. The test device uses the Introspect() method on the Introspectable interface to locate an ImageSink child bus object of the Stream bus object specified by the TestObjectPath test parameter.
   - If one is not found, execution of the test case ends.
   - If one is found, the test device calls the Open() method on the Stream bus object.
4. The test device retrieves the Capabilities property on the Stream bus object.
5. The test device calls the Connect() method on the ImageSink bus object with the following values for the parameters:
   - host: unique name of the test device's bus attachment
   - objectPath: "/Player/Out/Image"
   - configuration: first capability based in the list of capabilities retrieved from the Capabilities property,
6. The test device sends a Data signal with a zero-sized array to the ImageSink bus object.
7. The test device calls the Close() method on the Stream bus object.
8. The test device leaves the session.

Expected results

- If an ImageSink child bus object is not found, a note is added that the Stream does not support ImageSink.
- The Open() method calls return successfully without an error.
- The Capabilities property was successfully received without an error.
- The Connect() method calls return successfully without an error.
- A Data signal with a zero sized array can be emitted without an error.
- The Close() method calls return successfully without an error.

3.20 Audio-v1-20: Sending data to an Application.MetadataSink

Objective

Verify that a data signal can be emitted to an Application.MetadataSink.

Procedure

1. The test device listens for an About announcement from the application on the DUT.

2. After receiving an About Announcement from the application, the test device joins a session with the application at the port specified in the received About announcement.
3. The test device uses the Introspect() method on the Introspectable interface to locate an Application.MetadataSink child bus object of the Stream bus object specified by the TestObjectPath test parameter.
   - If one is not found, execution of the test case ends.
   - If one is found, the test device calls the Open() method on the Stream bus object.
4. The test device retrieves the Capabilities property on the Stream bus object.
5. The test device calls the Connect() method on the Application.MetadataSink bus object with the following values for the parameters:
   - host: unique name of the test device's bus attachment
   - objectPath: "/Player/Out/Audio"
   - configuration: first capability based in the list of capabilities retrieved from the Capabilities property.
6. The test device sends a Data signal with the following metadata keys and values to the Application.MetadataSink bus object:
   - Name: item name
   - Album: album title
7. The test device calls the Close() method on the Stream bus object.
8. The test device leaves the session.

Expected results

- If an Application.MetadataSink child bus object is not found, a note is added that the Stream does not support Application.MetadataSink.
- The Open() method calls return successfully without an error.
- The Capabilities property was successfully received without an error.
- The Connect() method calls return successfully without an error.
- A Data signal can be emitted without an error.
- The Close() method calls return successfully without an error.

3.21 Audio-v1-21: Setting the mute state on an AudioSink

Objective

Verify that an AudioSink can be muted/unmuted.

Procedure

1. The test device listens for an About announcement from the application on the DUT.

2. After receiving an About Announcement from the application, the test device joins a session with the application at the port specified in the received About announcement.
3. The test device uses the Introspect() method on the Introspectable interface to locate an AudioSink child bus object of the Stream bus object specified by the TestObjectPath test parameter.
   - If one is not found, execution of the test case ends.
   - If one is found, the test device calls the Open() method on the Stream bus object.
4. The test device calls the Connect() method on the AudioSink bus object with the following values for the parameters:
   - host: unique name of the test device's bus attachment
   - objectPath: "/Player/Out/Audio"
   - configuration: media type of "audio/x-raw" with the following parameters:
     - Channels: 1
     - Format: s16le
     - Rate: 44100
5. The test device retrieves the Muted property of the VolumeControl interface on the AudioSink bus object.
6. The test device sets the Muted property of the VolumeControl interface on the AudioSink bus object to its opposite value.
7. The test device waits to receive the MutedChanged signal.
8. The test device retrieves the Muted property of the VolumeControl interface from the AudioSink bus object.
9. The test device sets the Muted property of the VolumeControl interface on the AudioSink bus object to its opposite value.
10. The test device waits to receive the MutedChanged signal from the AudioSink bus object.
11. The test device calls the Close() method on the Stream bus object.
12. The test device leaves the session.

Expected results

- If an AudioSink child bus object is not found, a note is added that the Stream does not support AudioSink.
- The Open() method calls return successfully without an error.
- The Connect() method calls return successfully without an error.
- The Muted property can be retrieved.
- The Muted property can be set to its opposite value.

- After setting the value, the test device receives a MutedChanged signal with a muted value matching what was set.
- The Muted property can be retrieved again and it matches the value that was set.
- The Muted property can be set to its opposite value again.
- After setting the value, the test device receives a MutedChanged signal with a muted value matching what was set.
- The Muted property can be retrieved again and it matches the value that was set.
- The Close() method calls return successfully without an error.

3.22 Audio-v1-22: Setting the volume on an AudioSink

Objective

Verify that the volume can be set on an AudioSink.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About Announcement from the application, the test device joins a session with the application at the port specified in the received About announcement.
3. The test device uses the Introspect() method on the Introspectable interface to locate an AudioSink child bus object of the Stream bus object specified by the TestObjectPath test parameter.
   - If one is not found, execution of the test case ends.
   - If one is found, the test device calls the Open() method on the Stream bus object.
4. The test device calls the Connect() method on the AudioSink bus object with the following values for the parameters:
   - host: unique name of the test device’s bus attachment
   - objectPath: “/Player/Out/Audio”
   - configuration: media type of "audio/x-raw" with the following parameters:
     - Channels: 1
     - Format: s16le
     - Rate: 44100
5. The test device retrieves the VolumeRange property of the VolumeControl interface on the AudioSink bus object.
6. The test device retrieves the Volume property of the VolumeControl interface on the AudioSink bus object.

7. The test device sets the Volume property of the VolumeControl interface on the AudioSink bus object to a value that is one step value greater than the current value as long as the new value remains within the VolumeRange property.
8. If the volume is changed, the test device waits to receive the VolumeChanged signal from the AudioSink bus object.
9. The test device sets the Volume property of the VolumeControl interface on the AudioSink bus object to a value that is one step value less than the current value as long as the new value remains within the VolumeRange property.
10. If the volume is changed, the test device waits to receive the VolumeChanged signal from the AudioSink bus object.
11. The test device calls the Close() method on the Stream bus object.
12. The test device leaves the session.

Expected results

- If an AudioSink child bus object is not found, a note is added that the Stream does not support AudioSink.
- The Open() method calls return successfully without an error.
- The Connect() method calls return successfully without an error.
- The VolumeRange property can be retrieved. The following must be true of the VolumeRange values:
  - The low value must be less than the high value.
  - The step value must be greater than zero.
  - The difference between the high and low values must be divisible by the step value with no remainder.
- The Volume property can be retrieved and its current value is a valid step value within the VolumeRange.
- The Volume property can be set to one step value less than the current value (if the new value is within the VolumeRange).
- After setting the value, the test device receives a VolumeChanged signal with a volume value matching what was set.
- The Volume property can be set to one step value more than the current value (if the new value is within the VolumeRange).
- After setting the value, the test device receives a VolumeChanged signal with a volume value matching what was set.
- The Close() method calls return successfully without an error.

3.23 Audio-v1-23: Setting an invalid volume on an AudioSink

Objective

Verify that setting an invalid volume on an AudioSink will return an error.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About Announcement from the application, the test device joins a session with the application at the port specified in the received About announcement.
3. The test device uses the Introspect() method on the Introspectable interface to locate an AudioSink child bus object of the Stream bus object specified by the TestObjectPath test parameter.
   - If one is not found, execution of the test case ends.
   - If one is found, the test device calls the Open() method on the Stream bus object.
4. The test device calls the Connect() method on the AudioSink bus object with the following values for the parameters:
   - host: unique name of the test device's bus attachment
   - objectPath: "/Player/Out/Audio"
   - configuration: media type of "audio/x-raw" with the following parameters:
     - Channels: 1
     - Format: s16le
     - Rate: 44100
5. The test device retrieves the VolumeRange property of the VolumeControl interface on the AudioSink bus object.
6. If the minimum INT16 value (-32768) is not within the VolumeRange, the test device sets the Volume property of the VolumeControl interface on the AudioSink bus object to that value and waits for a second to see if a VolumeChanged signal is received.
7. If the maximum INT16 value (32767) is not within the VolumeRange, the test device sets the Volume property of the VolumeControl interface on the AudioSink bus object to that value and waits for a second to see if a VolumeChanged signal is received.
8. The test device calls the Close() method on the Stream bus object.
9. The test device leaves the session.

Expected results

- If an AudioSink child bus object is not found, a note is added that the Stream does not support AudioSink.
- The Open() method calls return successfully without an error.

- The Connect() method calls return successfully without an error.
- The VolumeRange property can be retrieved.
- If the minimum Volume value is greater than the minimum INT16 value, attempting to set the Volume property to the minimum INT16 value will result in an error and a VolumeChanged signal is not received.
- If the maximum Volume value is less than the maximum INT16 value, attempting to set the Volume property to the maximum INT16 value will result in an error and a VolumeChanged signal is not received.
- The Close() method calls return successfully without an error.

3.24 Audio-v1-24: Independence of mute and volume on an AudioSink

Objective

Verify that setting an invalid volume on an AudioSink will return an error.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About Announcement from the application, the test device joins a session with the application at the port specified in the received About announcement.
3. The test device uses the Introspect() method on the Introspectable interface to locate an AudioSink child bus object of the Stream bus object specified by the TestObjectPath test parameter.
   - If one is not found, execution of the test case ends.
   - If one is found, the test device calls the Open() method on the Stream bus object.
4. The test device calls the Connect() method on the AudioSink bus object with the following values for the parameters:
   - host: unique name of the test device’s bus attachment
   - objectPath: “/Player/Out/Audio”
   - configuration: media type of "audio/x-raw" with the following parameters:
     - Channels: 1
     - Format: s16le
     - Rate: 44100
5. The test device retrieves the VolumeRange property of the VolumeControl interface on the AudioSink bus object.
6. The test device sets the Volume property of the VolumeControl interface on the AudioSink bus object to the middle value in its range.

7. The test device retrieves the Volume property of the VolumeControl interface on the AudioSink bus object.
8. The test device sets the Muted property of the VolumeControl interface on the AudioSink bus object to true.
9. The test device retrieves the Muted property of the VolumeControl interface on the AudioSink bus object.
10. The test device retrieves the Volume property of the VolumeControl interface on the AudioSink bus object.
11. The test device sets the Muted property of the VolumeControl interface on the AudioSink bus object to false.
12. The test device retrieves the Muted property of the VolumeControl interface on the AudioSink bus object.
13. The test device retrieves the Volume property of the VolumeControl interface on the AudioSink bus object.
14. The test device sets the Volume property of the VolumeControl interface on the AudioSink bus object to one step value more than its minimum value.
15. The test device retrieves the Volume property of the VolumeControl interface on the AudioSink bus object.
16. The test device sets the Muted property of the VolumeControl interface on the AudioSink bus object to true.
17. The test device retrieves the Muted property of the VolumeControl interface on the AudioSink bus object.
18. The test device retrieves the Volume property of the VolumeControl interface on the AudioSink bus object.
19. The test device calls the Close() method on the Stream bus object.
20. The test device leaves the session.

Expected results

- If an AudioSink child bus object is not found, a note is added that the Stream does not support AudioSink.
- The Open() method calls return successfully without an error.
- The Connect() method calls return successfully without an error.
- The VolumeRange property can be retrieved.
- Setting the Muted property to true does not change the Volume property value.
- Setting the Muted property to false does not change the Volume property value.
- The Volume property retrieved is equal to the value of the Volume property set.
- The Muted property retrieved is equal to the value of the Muted property set.
- Setting the Volume property to a new value one step value more than the lowest allowed value does not change the Muted property value.

- The Close() method calls return successfully without an error.

3.25 Audio-v1-25: Synchronize clocks on an AudioSink

Objective

Verify that the clocks can be synchronized with an AudioSink supporting synchronization.

Procedure

1. The test device listens for an About announcement from the application on the DUT.
2. After receiving an About Announcement from the application, the test device joins a session with the application at the port specified in the received About announcement.
3. The test device uses the Introspect() method on the Introspectable interface to verify that the Stream bus object specified by the TestObjectPath test parameter supports the Stream.Clock interface and to locate an AudioSink child bus object.
   - If the Stream.Clock interface is not supported or an AudioSink child bus object is not found, execution of the test case ends.
   - If the Stream.Clock interface is supported or an AudioSink child bus object is found, the test device retrieves the Version property from the Stream.Clock interface.
4. The test device calls the Open() method on the Stream bus object.
5. The test device calls the Connect() method on the AudioSink bus object with the following values for the parameters:
   - host: unique name of the test device’s bus attachment
   - objectPath: “/Player/Out/Audio”
   - configuration: media type of "audio/x-raw" with the following parameters:
     - Channels: 1
     - Format: s16le
     - Rate: 44100
6. The test device calls the SetTime() method on the Stream bus object with the current time.
7. After receiving the reply, the test device calculates the time between calling the SetTime() method and when a reply is received.
8. The test device calls the AdjustTime() method on the Stream bus object with a value that represents half of the elapsed time between calling the SetTime() method and receiving the reply.
9. The test device calls the Close() method on the Stream bus object.
10. The test device leaves the session.

Expected results

- If an AudioSink child bus object is not found, a note is added that the Stream does not support AudioSink.
- If an AudioSink child bus object is found but the Stream does not support Stream.Clock, a note is added that the Stream does not support Stream.Clock.
- The Version property retrieved from the Stream.Clock interface on the Stream bus object is 1.
- The Open() method calls return successfully without an error.
- The Connect() method calls return successfully without an error.
- The SetTime() method call returns successfully without an error.
- The AdjustTime() method call returns successfully without an error.
- The Close() method calls return successfully without an error.