

audiocapturemgr

- [Overview](#)
- [Data Flow](#)
- [Test Case Using Test App CLI Interface](#)
- [API Documentation](#)

Overview

- Audio capture manager presents the audio data to registered applications.
- Supports creation of audio clips from currently streaming content.
- Creates clips of audio from content currently viewing so that it can be used for song identification service.
- The [Data Capture Service Manager API](#) is used to create audio clip and post/stream to Music ID server (specified URL.)
- When Service Manager API is called to create audio clip, settop creates audio clip of specified duration.

Data Flow

- At startup, app calls [org.rdk.dataCapture_1.enableAudioCapture](#) api to create pre-capture audio buffer and starts buffering primary audio stream.
- App makes request for audio clip capture by passing following parameters:
 - Destination URL and application parameters for Music ID service
 - Length of clip capture
 - Primary or secondary audio. (Note: Only primary audio is continuously captured)
 - Send pre-captured data, or post-capture (capture starts after app makes request). Majority of requests will be for pre-captured data, where entire clip is already captured by the time call is made.
- App calls Service Manager API [org.rdk.dataCapture_1.getAudioClip](#) and passes the request

```
{
  "stream" : "primary"
  "url" : "http://<musicid_server/analyze?trx=83cf6049-b722-4c44-b92e-79a504ae8f85:1458580048400&codec=PCM_16_16K&deviceId=5082732351093257712",
  "duration" : 6
  "captureMode" : "preCapture"
}
```

- Service Manager stores url and then calls IARMBUS_AUDIOCAPTUREMGR_REQUEST_SAMPLE which is handled in audiocapturemgr.
- ACM (using audio driver api's) extracts the data based on specified length in seconds,
 - the content must be raw binary data captured from PCM output
 - packages data in a pcm file, and
 - sends IARM event to Service Manager.
 - Name of event: DATA_CAPTURE_IARM_EVENT_AUDIO_CLIP_READY.
 - Associated data: file or html locator is sent with event.
- Service Manager takes the audio clip file, formats the curl message and sends the file to URL pointing to Music ID service.

```
curl -H "Content-Type: application/x-wav" --trace-ascii debugdump.txt --data-binary @AudioClip_6.wav
'<media_service_url>&codec=PCM_16_16K&deviceId=<device ID>'
```

- Service Manager logs the capture event
- Service Manager logs the response from Music ID service.

Test Case Using Test App CLI Interface

Ensure that SDK support is enabled for audiocapture.

1. Confirm the presence of /usr/bin/audiocapturemgrtestapp
2. Launch /usr/bin/audiocapturemgrtestapp and verify that you get a CLI menu
3. Select option set precapture length.
4. Provide 3 as the value and press enter.
5. Select option Start client.
6. Wait 3 seconds and then select dump precaptured sample. The console will indicate the name and path of the file being written. Note this down and the precapture length you set (3).
7. Select capture next N seconds. Enter 10 for length and press enter. The console will indicate the name and path of the file being written. Note this down and the length you entered just now.
8. Wait 11 seconds.
9. Select Stop client.
10. Select Quit.
11. Copy the dumped files to a PC and verify that they're playable.
12. Verify that the duration of these sound clips match the number that you logged against them when conducting the test.

13. Delete all the .wav files created in /opt/ by the test application.

Note: the precaptured sample files have the following pattern: /opt/precap-*.wav.

The sound samples captured using capture next N seconds option will have the following filename pattern: /opt/opt/freshcap-*.wav.

API Documentation

To see the List of APIs supported by audiocapturemgr, Please see [Audiocapturemgr API Documentation](#)