

Stream dump — developer debug recording (FR 0.1)

FR 0.1 stream dump while casting

Table of contents

[Goals](#)

[Modes 0.1.1 vs 0.1.2](#)

[Archive format](#)

[meta.json schema](#)

[Wire frame format \(raw files\)](#)

[Android implementation plan](#)

[Visualization / 0.1.3](#)

[Tests](#)

Source: 20260608_STREAM_DUMP_DEBUG.md — section links work in PDF viewers that support internal anchors.

Emergency debug — not an alpha release blocker. Enables 6.5 (audio lag) and 6.6 analysis.

Parent: 20260608_ALPHA_PRIORITIES.md §0.1

Goals

- Record **elementary** encoded streams during an active cast session (sender first; receiver optional later).
- Produce analyst-friendly bundles compatible with session stats tooling (0.1.3).
- Dev-only; default **off**; bounded disk (e.g. 100 MB or 5 min).

Modes 0.1.1 vs 0.1.2

Mode	ID	Content	Use
Structured	0.1.1	Raw annex-B / ADTS-style with rich meta.json	Preferred for graphs replay
Raw quick	0.1.2	Same files, minimal meta, no zip compression of streams	Fastest path to ship

Both use the **same archive naming**; only record_mode and optional fields differ.

Archive format

```
{filesDir}/stream_dumps/20260608_120422_sess_record.zip
meta.json
20260608_audio_stream.raw
20260608_video_stream.raw
```

Pull via adb pull or share intent from dev settings.

meta.json schema

Field	Type	Description
schema_version	int	1
session_id	string	yyyyMMdd_HHmss
role	string	sender \
record_mode	string	raw_elementary \
started_at_epoch_ms	long	
ended_at_epoch_ms	long	
video	object?	absent if audio-only
audio	object?	absent if video-only
cast_settings_snapshot	object	transport, protection, codecs

Per-stream object:

Field	Description
mime	e.g. video/avc, audio/mp4a-latm
file	basename inside zip
format	length_prefix — see below
frame_count	int
sample_rate / channels	audio
width / height	video
csd0_hex / csd1_hex	codec config (video)

Wire frame format (raw files)

Each frame in .raw:

[4 bytes BE length][payload bytes]

Payload = same bytes as CastProtocol audio/video frame body (after codec). PTS optional in meta sidecar or future v2 header.

Android implementation plan

Step	Component
1	AppPreferences.KEY_DEV_STREAM_DUMP
2	StreamDumpWriter — thread-safe append, rotate on session end
3	Hook CastSendPump or encoder callbacks when flag on

4	On ScreenCastService stop → finalize zip + meta.json
5	Dev settings UI: toggle + "Open dumps folder"

No new native deps if Java-only zip (java.util.zip).

Visualization / 0.1.3

Tool	Update
SessionStatsRecorder / graphs ingest	Accept meta.json + frame counts
Future offline decoder script	scripts/decode_stream_dump.py (TBD)

Tests

- StreamDumpWriterTest — length-prefix round-trip, zip contains expected names
- Manual: cast 30s with dump on → pull zip → verify frame_count > 0