

RCS file: /Users/implicit/Desktop/Source
Code/cvs_strings/beads/audiosync/main/audiosync.c,v
Working file: bdk/beads/audiosync/main/audiosync.c
head: 1.26
branch:
locks: strict
access list:
symbolic names:

BUILD_20060123: 1.26
BUILD_20050908: 1.26
BUILD_20050817: 1.26
BUILD_20050722: 1.26
BUILD_20050718: 1.26
BUILD_20050627: 1.26
BUILD_20050605: 1.26
TRAVIS_20050527: 1.26.0.22
dev_NewSchema-branch: 1.26.0.20
rd_GuiToolkit-branch: 1.26.0.18
ALL_20050422: 1.26
TRIO_20050418: 1.26
TRIO_20050404: 1.26
blackfin-branch: 1.26.0.16
TRIO-20041130: 1.26
RADKIT_GOLD_0042: 1.26.0.14
BEADS_SILVER_0056: 1.26
RADKIT_GOLD_0041: 1.26.0.12
BEADS_SILVER_0055: 1.26
RADKIT_GOLD_0040: 1.26.0.10
BEADS_SILVER_0054: 1.26
BANDON_20040413: 1.26.0.8
RADKIT_GOLD_0039: 1.26.0.6
BEADS_SILVER_0053: 1.26
BANDON_20040329: 1.26.0.4
RADKIT_GOLD_0038: 1.26.0.2
BEADS_SILVER_0052: 1.26
RADKIT_GOLD_0037: 1.25.0.52
BEADS_SILVER_0051: 1.25
RADKIT_GOLD_0036: 1.25.0.50
BEADS_SILVER_0050: 1.25
RADKIT_GOLD_0035: 1.25.0.48
BEADS_SILVER_0049: 1.25
RADKIT_GOLD_0034: 1.25.0.46
BANDON_20031224: 1.25.0.44
BANDON_20031219: 1.25.0.42

BANDON_20031214: 1.25.0.40
BANDON_20031204: 1.25.0.38
BANDON_20031103: 1.25.0.36
BANDON_20031024: 1.25.0.34
BANDON_20031014: 1.25.0.32
BEADS_SILVER_0048: 1.25
BEADS_SILVER_0047: 1.25
RADKIT_GOLD_0033: 1.25.0.30
BANDON_20030903: 1.25.0.28
BEADS_SILVER_0046: 1.25
RADKIT_GOLD_0032: 1.25.0.26
BEADS_SILVER_0045: 1.25
RADKIT_GOLD_0031: 1.25.0.24
BEADS_SILVER_0044: 1.25
RADKIT_GOLD_0030: 1.25.0.22
BANDON_20030815: 1.25.0.20
BEADS_SILVER_0042: 1.25
RADKIT_GOLD_0029: 1.25.0.18
BEADS_SILVER_0041: 1.25
RADKIT_GOLD_0028: 1.25.0.16
BEADS_SILVER_0040: 1.25
BEADS_SILVER_0039: 1.25
RADKIT_GOLD_0026: 1.25.0.14
BEADS_SILVER_0038: 1.25
RADKIT_GOLD_0025: 1.25.0.12
BEADS_SILVER_0037: 1.25
RADKIT_GOLD_0024: 1.25.0.10
BEADS_SILVER_0036: 1.25
RADKIT_GOLD_0023: 1.25.0.8
BEADS_SILVER_0035: 1.25
RADKIT_GOLD_0022: 1.25.0.6
BEADS_SILVER_0034: 1.25
RADKIT_GOLD_0021: 1.25.0.4
BEADS_SILVER_0033: 1.25
RADKIT_GOLD_0020: 1.25.0.2
BEADS_SILVER_0032: 1.25
RADKIT_GOLD_0019: 1.23.0.2
BEADS_SILVER_0031: 1.23
RADKIT_GOLD_0018: 1.22.0.4
BEADS_SILVER_0030: 1.22
RADKIT_GOLD_0017: 1.22.0.2
BEADS_SILVER_0029: 1.22
RADKIT_GOLD_0016: 1.21.0.2
BEADS_SILVER_0028: 1.21

RADKIT_GOLD_0015: 1.20.0.4
BEADS_SILVER_0027: 1.20
RADKIT_GOLD_0014: 1.20.0.2
BEADS_SILVER_0026: 1.20
RADKIT_GOLD_0013: 1.18.0.6
BEADS_SILVER_0025: 1.18
RADKIT_GOLD_0012: 1.18.0.4
BEADS_SILVER_0024: 1.18
RADKIT_GOLD_0011: 1.18.0.2
BEADS_SILVER_0023: 1.18
RADKIT_GOLD_0010_INTERNAL: 1.18
BEADS_SILVER_0022: 1.18
RADKIT_GOLD_0009: 1.16.0.10
BEADS_SILVER_0021: 1.16
BEADS_SILVER_0020: 1.16
RADKIT_GOLD_0008_INTERNAL: 1.16
BEADS_SILVER_0019: 1.16
RADKIT_GOLD_0007: 1.16.0.8
BEADS_SILVER_0018: 1.16
BEADS_SILVER_0017: 1.16
RADKIT_GOLD_0006: 1.16.0.6
BEADS_SILVER_0016: 1.16
RADKIT_GOLD_0005_INTERNAL: 1.16
BEADS_SILVER_0015: 1.16
RADKIT_GOLD_0004_INTERNAL: 1.16
BEADS_SILVER_0014: 1.16
RADKIT_GOLD_0003_INTERNAL: 1.16
BEADS_SILVER_0013: 1.16
RADKIT_GOLD_0002: 1.16.0.4
BEADS_SILVER_0012: 1.16
BEADS_SILVER_0011: 1.16
RADKIT_GOLD_0001: 1.16.0.2
BEADS_SILVER_0010: 1.16
BEADS_SILVER_0009: 1.16
BEADS_SILVER_0008: 1.12
BEADS_SILVER_0007: 1.8
SILVER: 1.26

keyword substitution: kv

total revisions: 26; selected revisions: 26

description:

revision 1.26

date: 2004-03-04 18:23:22 -0600; author: build; state: Exp; lines: +81 -83;

Fix GCC type-punning warnings, which show up in retail mode on GCC 3.3.2.

Fix some coding convention violations (mostly whitespace).

revision 1.25

date: 2002-06-10 16:45:14 -0500; author: davidc; state: Exp; lines: +5 -91;
Removed the "preroll" edge, which is not used and whose purpose is unknown.

revision 1.24

date: 2002-06-10 16:18:52 -0500; author: davidc; state: Exp; lines: +77 -53;
Changed the default "smoothing" parameter to 1, so that the sliding average only includes the current value. Changed the code to silence all audio while the master clock is not set.
Changed the code from silencing the first audio message to sending an empty message before sending the first message, which allows the render clock to be updated without any data loss.

These changes help lessen the severity of audio chop that is heard on slave paths when the master is on a different machine than the source.

The problem was that the delay for each audio chunk is the average of the last eight audio chunks played. This "smoothing" technique caused the master/render clock difference to oscillate while converging, which caused chopping. To explain: when audio is early, AudioSync delays the audio. When AudioSync has delayed the audio to make the audio on-time, the sliding average makes AudioSync act as if the audio is still early, causing it to further delay the audio. This makes the audio late. The audio did converge to being on-time, but not before sliding from early to on-time to late to on-time. Because AudioSync silences the audio when it is early or late and plays it when it is on-time, this oscillation causes audio to be silenced, then played, then silenced, then played, which is the chopping.

The end result of these changes is that synchronization converge is sped up slightly and the out-of-sync slaves are silent instead of chopping. In the worst case, when the master is on a different machine than the source, this can cause between one and two seconds of silence.

This is more of a work-around than a fix for the real problem, which is that it takes time for updates to the render clock of the master path to be propagated to the slaves. I don't know of a good way to solve this problem.

revision 1.23

date: 2002-06-10 12:04:52 -0500; author: davidc; state: Exp; lines: +22 -7;
Fixed a memory overwrite bug in SlidingAvg_Init. Memory would be overwritten if the sliding average length was less than five.

revision 1.22

date: 2002-05-24 18:46:48 -0500; author: davidc; state: Exp; lines: +150 -148;
Made it so that the "MasterClockOffset" attribute is treated as a signed value, so the audio can be sped-up in addition to being slowed down. Made it so that "MasterClockOffset" is ignored in master paths.

Refactored AudioSync_Adjust so that it can be called multiple times for a single message. This was useful when playing around with different strategies of how to incorporate the MasterClockOffset. I did not end up calling AudioSunc_Adjust from more than one place, but I kept the changes to make it easier to play around with the MasterClockOffset strategy in the future.

revision 1.21

date: 2002-05-16 22:26:03 -0500; author: davidc; state: Exp; lines: +4 -4;
Untabified

revision 1.20

date: 2002-04-30 21:55:17 -0500; author: davidc; state: Exp; lines: +60 -42;
Added some code that reads in the "MasterClockOffset" from the path and adds that the to master clock's epoch. This hack helps simpletv synchronize audio and video.

revision 1.19

date: 2002-04-30 20:11:26 -0500; author: davidc; state: Exp; lines: +19 -16;
Fixed defect #257, which is that audiosync would go into an inifite loop when it received an empty audio package that was within the EARLY_COPY range. I fixed this by not attempting to duplicate empty audio messages.

revision 1.18

date: 2002-04-03 15:15:46 -0600; author: davidc; state: Exp; lines: +2 -2;
Made the "Audio sync reports" DEBUGOUT a data trace (level 4) rather than a detail (level 5) so that it can be turned on without turning the others on.

revision 1.17

date: 2002-04-02 23:23:52 -0600; author: davidc; state: Exp; lines: +3 -1;
Added static keyword to eliminate unnecessary namespace pollution.

revision 1.16

date: 2001-11-09 18:53:11 -0600; author: guyc; state: Exp; lines: +1 -26;
Removed old and configuration constants.

revision 1.15

date: 2001-11-09 16:52:01 -0600; author: guyc; state: Exp; lines: +11 -11;
Changed capitalization of path parameters.

revision 1.14

date: 2001-11-09 16:49:04 -0600; author: guyc; state: Exp; lines: +120 -20;
Now optionally takes parameters (lots of them) from the path context. Default values have changed back to favour faster convergence, which may not work will with

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.