```
RCS file: /Users/implicit/Desktop/Source Code/cvs_strings/beads/timesync/main/clocksync.c,v
Working file: bdk/beads/timesync/main/clocksync.c
head: 1.32
branch:
locks: strict
access list:
symbolic names:
	BUILD_20060123: 1.32
	BUILD_20050908: 1.32
	BUILD_20050817: 1.32
	BUILD_20050722: 1.32
	BUILD_20050718: 1.32
	BUILD_20050627: 1.32
	BUILD_20050605: 1.32
	TRAVIS_20050527: 1.32.0.22
	dev_NewSchema-branch: 1.32.0.20
	rd_GuiToolkit-branch: 1.32.0.18
	ALL_20050422: 1.32
	TRIO_20050418: 1.32
	TRIO_20050404: 1.32
	blackfin-branch: 1.32.0.16
	TRIO-20041130: 1.32
	RADKIT_GOLD_0042: 1.32.0.14
	BEADS_SILVER_0056: 1.32
	RADKIT_GOLD_0041: 1.32.0.12
	BEADS_SILVER_0055: 1.32
	RADKIT_GOLD_0040: 1.32.0.10
	BEADS_SILVER_0054: 1.32
	BANDON_20040413: 1.32.0.8
	RADKIT_GOLD_0039: 1.32.0.6
	BEADS_SILVER_0053: 1.32
	BANDON_20040329: 1.32.0.4
	RADKIT_GOLD_0038: 1.32.0.2
	BEADS_SILVER_0052: 1.32
	RADKIT_GOLD_0037: 1.31.0.52
	BEADS_SILVER_0051: 1.31
	RADKIT_GOLD_0036: 1.31.0.50
	BEADS_SILVER_0050: 1.31
	RADKIT_GOLD_0035: 1.31.0.48
	BEADS_SILVER_0049: 1.31
	RADKIT_GOLD_0034: 1.31.0.46
	BANDON_20031224: 1.31.0.44
	BANDON_20031219: 1.31.0.42
	BANDON_20031214: 1.31.0.40
```

```
BANDON_20031204: 1.31.0.38
BANDON_20031103: 1.31.0.36
BANDON_20031024: 1.31.0.34
BANDON_20031014: 1.31.0.32
BEADS_SILVER_0048: 1.31
BEADS_SILVER_0047: 1.31
RADKIT_GOLD_0033: 1.31.0.30
BANDON_20030903: 1.31.0.28
BEADS_SILVER_0046: 1.31
RADKIT_GOLD_0032: 1.31.0.26
BEADS_SILVER_0045: 1.31
RADKIT_GOLD_0031: 1.31.0.24
BEADS_SILVER_0044: 1.31
RADKIT_GOLD_0030: 1.31.0.22
BANDON_20030815: 1.31.0.20
BEADS_SILVER_0042: 1.31
RADKIT_GOLD_0029: 1.31.0.18
BEADS_SILVER_0041: 1.31
RADKIT_GOLD_0028: 1.31.0.16
BEADS_SILVER_0040: 1.31
BEADS_SILVER_0039: 1.31
RADKIT_GOLD_0026: 1.31.0.14
BEADS_SILVER_0038: 1.31
RADKIT_GOLD_0025: 1.31.0.12
BEADS_SILVER_0037: 1.31
RADKIT_GOLD_0024: 1.31.0.10
BEADS_SILVER_0036: 1.31
RADKIT_GOLD_0023: 1.31.0.8
BEADS_SILVER_0035: 1.31
RADKIT_GOLD_0022: 1.31.0.6
BEADS_SILVER_0034: 1.31
RADKIT_GOLD_0021: 1.31.0.4
BEADS_SILVER_0033: 1.31
RADKIT_GOLD_0020: 1.31.0.2
BEADS_SILVER_0032: 1.31
RADKIT_GOLD_0019: 1.30.0.4
BEADS_SILVER_0031: 1.30
RADKIT_GOLD_0018: 1.30.0.2
BEADS_SILVER_0030: 1.30
RADKIT_GOLD_0017: 1.29.0.2
BEADS_SILVER_0029: 1.29
RADKIT_GOLD_0016: 1.28.0.2
BEADS_SILVER_0028: 1.28
RADKIT_GOLD_0015: 1.26.0.2
```

```
	BEADS_SILVER_0027: 1.26
	RADKIT_GOLD_0014: 1.18.0.2
	BEADS_SILVER_0026: 1.18
	RADKIT_GOLD_0013: 1.17.0.2
	BEADS_SILVER_0025: 1.17
	RADKIT_GOLD_0012: 1.16.0.4
	BEADS_SILVER_0024: 1.16
	RADKIT_GOLD_0011: 1.16.0.2
	BEADS_SILVER_0023: 1.16
	RADKIT_GOLD_0010_INTERNAL: 1.16
	BEADS_SILVER_0022: 1.16
	RADKIT_GOLD_0009: 1.14.0.6
	BEADS_SILVER_0021: 1.14
	BEADS_SILVER_0020: 1.14
	RADKIT_GOLD_0008_INTERNAL: 1.14
	BEADS_SILVER_0019: 1.14
	RADKIT_GOLD_0007: 1.14.0.4
	BEADS_SILVER_0018: 1.14
	BEADS_SILVER_0017: 1.14
	RADKIT_GOLD_0006: 1.14.0.2
	BEADS_SILVER_0016: 1.14
	RADKIT_GOLD_0005_INTERNAL: 1.14
	BEADS_SILVER_0015: 1.14
	RADKIT_GOLD_0004_INTERNAL: 1.14
	BEADS_SILVER_0014: 1.14
	RADKIT_GOLD_0003_INTERNAL: 1.14
	BEADS_SILVER_0013: 1.14
	RADKIT_GOLD_0002: 1.13.0.4
	BEADS_SILVER_0012: 1.13
	BEADS_SILVER_0011: 1.13
	RADKIT_GOLD_0001: 1.13.0.2
	BEADS_SILVER_0010: 1.13
	BEADS_SILVER_0009: 1.12
	BEADS_SILVER_0008: 1.11
	BEADS_SILVER_0007: 1.8
	BEADS_SILVER_0006: 1.4
	SILVER: 1.32
keyword substitution: kv
total revisions: 32;	selected revisions: 32
description:
----------------------------
revision 1.32
date: 2004-03-09 21:45:51 -0600;  author: build;  state: Exp;  lines: +60 -131;
Fix GCC type-punning warnings, which are new as of GCC 3.3.2.
Fix coding convention violations (mostly whitespace)
----------------------------
revision 1.31
date: 2002-06-10 15:30:07 -0500;  author: davidc;  state: Exp;  lines: +22 -2;
Added a printout for when the epoch of a remote render clock, which is probably a master
clock, changes by more than five milliseconds, since this has a noticable effect on
synchornization.
----------------------------
revision 1.30
date: 2002-05-31 15:01:28 -0500;  author: davidc;  state: Exp;  lines: +7 -5;
Added the value of the master clock epoch to the "Encoding Master Clock Epoch" DEBUGOUT.
----------------------------
revision 1.29
date: 2002-05-24 18:38:54 -0500;  author: davidc;  state: Exp;  lines: +5 -5;
Made it so that the MasterClockOffset is treated as a signed value.
----------------------------
revision 1.28
date: 2002-05-17 14:50:30 -0500;  author: davidc;  state: Exp;  lines: +4 -2;
Added the value of the master clock offset to the SOS_DEBUGOUT_DETAIL that said when the
decode edge received a master clock offset.
----------------------------
revision 1.27
date: 2002-05-17 14:24:20 -0500;  author: davidc;  state: Exp;  lines: +84 -12;
Added code to send updates to the "MasterClockOffset" path attribute from the source to the
sink.
----------------------------
revision 1.26
date: 2002-05-10 11:54:03 -0500;  author: davidc;  state: Exp;  lines: +2 -102;
Removed the deprecated ClockSync[MasterEncode] and ClockSync[SlaveEncode] edges.
----------------------------
revision 1.25
date: 2002-05-09 17:21:54 -0500;  author: davidc;  state: Exp;  lines: +59 -45;
Added a new flag, FLAG_ISMASTERPATHISVALID, which determines if the FLAG_ISMASTERPATH
should be ignored or not.  This makes it so that loopback packets don't have to set
FLAG_ISMASTERPATH, which eliminates the risk of the sink changing the master-ness state on
the source.
----------------------------
revision 1.24
date: 2002-05-09 13:44:00 -0500;  author: davidc;  state: Exp;  lines: +8 -2;
Added code to set the FLAG_ISMASTERPATH flag when the Encode edge detects that it belongs
to the master path.
----------------------------
revision 1.23
date: 2002-05-09 12:35:57 -0500;  author: davidc;  state: Exp;  lines: +4 -9;
```

```
Changed the routine that determines the master-ness of the current path to use
SOS_ISAMPLECLOCK::IsSameAs, instead of a simple pointer comparison.
----------------------------
revision 1.22
date: 2002-05-09 12:28:34 -0500;  author: davidc;  state: Exp;  lines: +24 -7;
Added comments.
----------------------------
revision 1.21
date: 2002-05-09 00:48:38 -0500;  author: davidc;  state: Exp;  lines: +172 -18;
Changed the "Encode" edge to send master clock updates forward iff it's in a slave path and to
request that render clock updates get sent backward iff it's in a master path.

Changed the "Decode" edge to set the master clock to refer to the render clock if it's in a path
and to set the master clock to refer to a different clock if it's in a slave path.

The main is that, for all master sinks, the master clock and the render clock refer to the same
object.  This eliminates the latency between when the master sink updates its render clock and
when its master clock is updated.  It also fixes a performance inconsistency between when the
master sink is on a different machine than the source and when the master sink is on the same
machine as the source.
----------------------------
revision 1.20
date: 2002-05-08 23:27:15 -0500;  author: davidc;  state: Exp;  lines: +217 -170;
Added the "Encode" edge, which will eventually take the place of both the "MasterEncode"
edge and the "SlaveEncode" edge.  It is currently just a place holder and is functionally
equivalent to the "SlaveEncode" edge.

Reorganized the way edges are registered to use a table-driven method.  This simplifies the
error handling in the SOS_BEAD_INITIALIZE routine.

Moved the logic that reads the clocks from the path to the path init routines.  This makes is so
that the message handlers can assume that the clocks have already been read.  (Before, this
logic was in the path init routines and the message handlers).

Removed some dead code, including the timer code that sends on the loopback and the
"RenderNeeded" boolean.
----------------------------
revision 1.19
date: 2002-05-08 22:04:05 -0500;  author: davidc;  state: Exp;  lines: +5 -5;
Renamed the "Encode" edge of the ClockSync bead to "SlaveEncode".

This is the first phase in merging the functionality of the "Encode" and "MasterEncode" edge.
This will allow me to add the new edge as the "Encode" edge without having to port every
package to use the new edge.
```

DOCKET ALARM

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.

WHAT WILL YOU BUILD? | sales@docketalarm.com | 1-866-77-FASTCASE

fastcase®
Smarter legal research.