

CONTAINS PROTECTIVE ORDER MATERIAL

U.S. Patent No. 7,391,791	Application of Claim Language to Source Code
Claim 1	
[1 Preamble] A method for synchronizing a rendering of a content provided by a source at one or more devices which are nodes of a network, the content having a rendering time, the method comprising:	<p>Source Code Folders:</p> <pre>SOFTWARE_STRUCTURE = \test\demo\rules\ SOFTWARE_CODE = \beads\</pre> <p>The Implicit Source Code specifies a distributed system consisting of devices that are nodes of a network (“devices”). These devices execute synchronizing rendering of content provided by a source, where each device has a rendering time corresponding to the device time when the device renders content.</p> <p>An architecture of one such distributed system comprising a plurality of devices defined in files <code>videomulti.rule</code>, <code>videoclient.rule</code>, <code>ipacmaudioserver.rule</code>, <code>syncaudio.rule</code>, and <code>timesync.rule</code>.</p>
[1a] designating one of the one or more devices as a master device, the master device having a	The distributed system architecture described in file <code>videomulti.rule</code> designates one of the network devices as a master device having a master device clock that provides the master rendering times. ^{4,5}

¹ These files are contained in folder `SOFTWARE_STRUCTURE\`

² Another similar distributed system that renders synchronized content streams is described by the source code in folder `\test\audiosync\`

³ See file `SOFTWARE_STRUCTURE\audiosync\package\package\timesync.rule`

⁴ Defined at lines 6 to 133 in file `SOFTWARE_STRUCTURE\videomulti.rule`

⁵ See lines 10 to 15 in file `SOFTWARE_STRUCTURE\videomulti.rule`

CONTAINS PROTECTIVE ORDER MATERIAL

U.S. Patent No. 7,391,791	Application of Claim Language to Source Code
<p>master device time and a master rendering time;</p>	<p>This master rendering device receives a combined audio and video content stream by a source device.⁶ Upon receiving the combined audio and video content stream, the master rendering device uses the <code>avidemux</code> bead to separate the combined audio and video stream into separate audio and video streams.^{7,8} The <code>avidemux</code> bead instantiates a master rendering clock <code>IAudioClock</code> associated with the combined audio and video content stream.⁹ This master rendering clock <code>IAudioClock</code> generates master rendering times that are indicative of the status of the rendering of the combined audio and video content stream by the master rendering device at the master rendering times.</p> <p>The Implicit Source Code implements a distributed system that runs on multiple devices including master rendering devices and slave rendering devices for synchronized audio and video rendering by master and slave rendering devices. The Implicit Source Code specifies a distributed system architecture in file <code>videomulti.rule</code>. Within file <code>videomulti.rule</code>, a master rendering device is defined to have a master rendering clock that provides master rendering times corresponding to when the master renders media content.¹⁰ This master rendering device is set up to receive combined audio and video content stream from a source device. The master rendering device receives a combined audio and video content stream</p>

⁶ See lines 7 to 79 in file `\test\demo\source.pl`

⁷ See lines 11 to 15 in file `SOFTWARE_STRUCTURE\videomulti.rule`

⁸ See file `SOFTWARE_CODE\avidemux\main\avidemux.c`

⁹ See lines 447 to 451 in file `SOFTWARE_CODE\avidemux\main\avidemux.c`

¹⁰ See lines 11 to 15 in file `SOFTWARE_STRUCTURE\videomulti.rule`

¹¹ See line 9 in file `SOFTWARE_STRUCTURE\videomulti.rule`

CONTAINS PROTECTIVE ORDER MATERIAL

U.S. Patent No. 7,391,791	Application of Claim Language to Source Code
	<p>local network port 8013 from a content source device.¹² As part of the process, this master rendering device uses the <code>avidemux</code> bead to split audio and video content stream into separate audio and video streams using functions <code>AvidDemux_EncodeMessageHandler</code>¹³ and <code>ChunkP</code>. The <code>avidemux</code> bead <code>avidemux</code> separate the combined audio and video content stream into separate audio and video content streams. In processing the combined audio and video content streams, the <code>avidemux</code> bead uses the function <code>AudioPrepare</code>.¹⁵ Function <code>AudioPrepare</code> initializes a master rendering clock <code>IAudioClock</code> associated with the audio content stream. The master rendering clock <code>IAudioClock</code> generates a plurality of master device time domains that are indicative of how much rendering of the combined audio and video content stream has occurred at the master rendering device at different times.</p> <p>The master rendering device also contains a master device time that is used to timestamp packets that the master rendering device transmits over the IP network. The master device time is used by the <code>timesync</code> bead to determine the time difference between the master and the slave time domains.^{19,20} The <code>timesync</code></p>

¹² See line 9 in file `SOFTWARE_STRUCTURE\videomulti.rule`

¹³ Implemented at lines 883 to 964 in file `SOFTWARE_CODE\avidemux\main\avidemux.c`

¹⁴ Implemented at lines 597 to 849 in file `SOFTWARE_CODE\avidemux\main\avidemux.c`

¹⁵ Implemented at lines 389 to 477 in file `SOFTWARE_CODE\avidemux\main\avidemux.c`

¹⁶ See lines 447 to 451 in file `SOFTWARE_CODE\avidemux\main\avidemux.c`

¹⁷ See lines 40 to 44 in file `SOFTWARE_STRUCTURE\videomulti.rule`

¹⁸ Defined at lines 16 to 26 in file `SOFTWARE_STRUCTURE\timesync.rule`

¹⁹ Defined at lines 5 to 13 in file `SOFTWARE_STRUCTURE\timesync.rule`

²⁰ See file `SOFTWARE_CODE\timesync\main\timesync.c`

CONTAINS PROTECTIVE ORDER MATERIAL

U.S. Patent No. 7,391,791	Application of Claim Language to Source Code
	<p>a function <code>TimeSync_Send</code> that periodically broadcasts the device times to the network.²¹ Function <code>TimeSync_Send</code> calls function <code>SOS_Clock_Get</code> to obtain the current device times.²² When the <code>timesync</code> bead runs on the rendering device, these times correspond to master device times. The <code>timesync</code> bead also implements a function <code>TimeSync_UpdateHandler</code>²³ that periodically sends packets sent to port 9123. Function <code>TimeSync_UpdateHandler</code> calls function <code>SOS_Clock_TickGet</code> to obtain the device time when the device sends a packet.²⁴ When the <code>timesync</code> bead is running on the master rendering device, a call to function <code>SOS_Clock_TickGet</code> provides the master device time. Afterwards, function <code>TimeSync_UpdateHandler</code> calls function <code>TimeSync_HostUpdate</code>^{25, 26} to determine the time differential between the master and the slave time domains using the master and slave device times.</p>
[1b] designating remaining devices among one of the one or more devices as at least one slave device, the at least one slave device	<p>The distributed system described in file <code>videomulti.rule</code>, <code>videoclient.rule</code>, and <code>syncaudio.rule</code> designates a plurality of devices as slave devices, each slave device has a slave device time and a master rendering time. The slave devices render decoded RGB video frames in synchronization with the master rendering device.</p>

²¹ Implemented at lines 519 to 603 in file `SOFTWARE_CODE\timesync\main\timesync.c`

²² See line 588 in file `SOFTWARE_CODE\timesync\main\timesync.c`

²³ Implemented at lines 737 to 854 in file `SOFTWARE_CODE\timesync\main\timesync.c`

²⁴ See line 745 in file `SOFTWARE_CODE\timesync\main\timesync.c`

²⁵ See lines 798 to 803 in file `SOFTWARE_CODE\timesync\main\timesync.c`

²⁶ Implemented at lines 225 to 358 in file `SOFTWARE_CODE\timesync\main\timesync.c`

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.