

```
1  /*+++++
...  +++++
2
3  Copyright (c) 2001 BeComm Corporation
4
5  Filename:
6
7      avidemux.c
8
9  Group Overview:
10
11     Extracts the incoming avi data stream into audio and
12     video.  Video is sent along the main path, while audio
13     is forked onto a new child path.
14
15  Owner:
16
17     Guy Carpenter (guyc) 17-Jul-2001
18
19  -----*/
...  ----*/
20
21  #define SOS_DEBUG_ZONE "/beads/avidemux"
22
23  #include <sosstrings.h>
24  #include <sosmultimedia.h>
25  #include "riffparser.h"
26  #include "aviformat.h"
27  #include "wavformat.h"
28  #include "bmpformat.h"
29
30  SOS_SOURCE_VERSION (
31      "$Id: avidemux.c,v 1.36 2001/10/02 21:07:24 guyc Exp $"
32  );
33
34  #define BEAD_NAME "avidemux"
35
36  /*
37   * Name of audio context to use if we get a pcm audio stream
38   */
39  #define AUDIOCONTEXT_CLASS "pcmcontext"
40  /*
41   * Name of video context to use if we get a bmp audio stream.
42   */
43  #define VIDEOCONTEXT_CLASS "vbecontext"
```

```
44 /*
45  * Name of sample clock class
46  */
47 #define SAMPLECLOCK_CLASS "sampleclock"
48
49 /*
50  * Specify level of detail in debug prints, from
51  * minimum to maximum the useful values are:
52  * 0 : no output
53  * SOS_DEBUG_LEVEL_FuncTrace : basic procedure entry logging
54  * SOS_DEBUG_LEVEL_DataTrace : data format logging
55  * SOS_DEBUG_LEVEL_Detail : per-chunk logging
56  */
57 #define DEBUG_LEVEL SOS_DEBUG_LEVEL_DataTrace
58
59 /*+++++
...
60 Configuration
61 +++++*/
62
63 /*
64  * The following parameters should really be set through
65  * the InitContext.
66  */
67
68 /*
69  * Maximum queue size (in messages) of the audio path
70  */
71 #define AUDIO_MAX_QUEUE_LENGTH "50"
72 /*
73  * play only 1 in N frame - useful on slow machines.
74  */
75 #define VIDEO_DECIMATE 1 /* 1 means play every frame */
76
77 /*+++++
...
78 Structures
79 +++++*/
80
81 /*
82  * AVIDEMUX_CONTEXT is the session context for the bead. Each
83  * active session of this bead will get it's own copy of this
84  * structure
```

```
85  */
86  typedef struct _AVIDEMUX_CONTEXT {
87      RIFF_CONTEXT *      Riff;          /* riff parser          */
88      RIFF_CHUNK *       Chunk;         /* current riff chunk  */
89      SOS_UINT32         StreamIndex;   /* counts streams in avi */
90
91      SOS_UINT32         VidStreamId;   /* fourcc identifier   */
92      SOS_UINT32         VidStreamIndex; /* index of video stream */
93      SOS_UINT32         VidFrameNumber; /* count of sent frames */
94      BMP_FORMAT         VidStreamFormat; /* bmp format header   */
95      SOS_UINT32         VidFrameRate;  /* frame rate numerator */
96      SOS_UINT32         VidFrameRateDivisor; /* frame rate div     */
97      SOS_IVIDEOCONTEXT * IVideoContext; /* propagates format   */
98
99      SOS_PATH *         AudioPath;     /* path for audio output */
100     SOS_UINT32         AudStreamId;    /* fourcc identifier     */
101     SOS_UINT32         AudStreamIndex; /* index of audio stream */
102     WAV_FORMAT         AudStreamFormat; /* audio format from wav */
103     SOS_IAUDIOCONTEXT * IAudioContext; /* propagates format     */
104
105     SOS_ISAMPLECLOCK * IAudioClock;    /* master sample clock  */
106     SOS_ISAMPLECLOCK * IVideoClock;    /* slave sample clock   */
107
108     SOS_UINT32         AudioSampleCount; /* total samples sent   */
109     SOS_UINT32         AudioSampleBytes; /* bytes per sample     */
110 } AVIDEMUX_CONTEXT;
111
112 /*+++++
...
113 Debug Output Functions
114 +++++*/
...
115
116 /*
117  * Print out the avi header for debugging
118  */
119 static
120 void
121 DumpAviHeader(
122     AVI_HEADER *   AviHeader
123 )
124 {
125     SOS_DEBUGOUT_DATA_TRACE(
126         "MicroSecPerFrame      %u\n"
127         "MaxBytesPerSec          %u\n"
```

```
128     "Reserved1          %u\n"
129     "Flags              %u\n"
130     "TotalFrames        %u\n"
131     "InitialFrames      %u\n"
132     "Streams            %u\n"
133     "SuggestedBufferSize %u\n"
134     "Width               %u\n"
135     "Height              %u\n"
136     "Scale               %u\n"
137     "Rate                %u\n"
138     "Start               %u\n"
139     "Length              %u\n\n",
140     AviHeader->MicroSecPerFrame,
141     AviHeader->MaxBytesPerSec,
142     AviHeader->Reserved1,
143     AviHeader->Flags,
144     AviHeader->TotalFrames,
145     AviHeader->InitialFrames,
146     AviHeader->Streams,
147     AviHeader->SuggestedBufferSize,
148     AviHeader->Width,
149     AviHeader->Height,
150     AviHeader->Scale,
151     AviHeader->Rate,
152     AviHeader->Start,
153     AviHeader->Length
154 );
155 }
156
157 /*
158  * Print out the avi stream header for debugging
159  */
160 static
161 void
162 DumpAviStreamHeader(
163     AVI_STREAMHEADER *AviStreamHeader
164 )
165 {
166     SOS_DEBUGOUT_DATA_TRACE(
167         "Type          %.4s [%lx]\n"
168         "Handler         %.4s\n"
169         "Flags            %u\n"
170         "Reserved1        %u\n"
171         "InitialFrames    %u\n"
172         "Scale            %u\n"
```

```
173     "Rate                %u\n"
174     "Length              %u\n"
175     "SuggestedBufferSize %u\n"
176     "Quality              %u\n"
177     "SampleSize          %u\n"
178     "Reserved2           %u\n"
179     "Reserved3           %u\n\n",
180     &AviStreamHeader->Type,
181     AviStreamHeader->Type,
182     &AviStreamHeader->Handler,
183     AviStreamHeader->Flags,
184     AviStreamHeader->Reserved1,
185     AviStreamHeader->InitialFrames,
186     AviStreamHeader->Scale,
187     AviStreamHeader->Rate,
188     AviStreamHeader->Start,
189     AviStreamHeader->Length,
190     AviStreamHeader->SuggestedBufferSize,
191     AviStreamHeader->Quality,
192     AviStreamHeader->SampleSize,
193     AviStreamHeader->Reserved2,
194     AviStreamHeader->Reserved3
195 );
196 }
197
198 /*
199  * Print out the wav format header for debugging
200  */
201 static
202 void
203 DumpWavFormat(
204     WAV_FORMAT *    WavFormat
205 )
206 {
207
208     SOS_DEBUGOUT_DATA_TRACE(
209         "WAV Header:\n"
210         "FormatTag          %u\n"
211         "Channels            %u\n"
212         "SamplesPerSec       %lu\n"
213         "AvgBytesPerSec      %lu\n"
214         "BlockAlign          %u\n"
215         "BitsPerSample       %u\n"
216         "Size                %u\n\n",
217         WavFormat->FormatTag,
```

# 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.