```
/*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++

Copyright (c) 2001 BeComm Corporation

Filename:

    framer.c

Group Name:

    todo

Group Overview:

    todo

Owner:

    Guy Carpenter (guyc) 16-Aug-2001

--------------------------------------------------------------------------------
---*/
#define DEBUG_ZONE "/beads/framer"

#include <sosstrings.h>

#include <stdio.h>

SOS_SOURCE_VERSION (
    "$Id: framer.c,v 1.3 2001/10/11 17:32:53 guyc Exp $"
);
/*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++
Named Constants
--------------------------------------------------------------------------------
---*/

/*
 * Name of bead
 */
static const char BEAD_NAME[]   = "framer";

/*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
41 Definitions
42 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
… +++*/
43
44 /*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
… +++++
45 globals
46 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
… +++*/
47
48 /*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
… +++++
49 Macros
50 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
… +++*/
51
52 /*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
… +++++
53 Types
54 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
… +++*/
55
56 typedef struct _FRAMER {
57     size_t         Length;
58     SOS_MESSAGE * Remainder;
59 } FRAMER;
60
61 /*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
… +++++
62 Helper Functions
63 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
… +++*/
64
65 /*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
… +++++
66 Context create and destory
67 --------------------------------------------------------------------------------
… ---*/
68
69 /*++
70 Routine Name:
71
72     Framer_ContextCreate
73
74 Routine Description:
```

```
75
76      Attempt to allocate a Framer session and all the resources it
…   requires.
77
78  Parameters:
79
80      None.
81
82  Return Value:
83
84      FRAMER * -
85          A valid pointer to a session context on success.
86
87          A NULL pointer if an error occured.
88
89  --*/
90  static
91  FRAMER *
92  Framer_ContextCreate(
93      void
94  )
95  {
96      FRAMER *context;
97
98      SOS_DEBUGOUT_FUNC_TRACE("Framer_ContextCreate called\n");
99
100     /* allocate memory for Framer session context */
101     context = SOS_Mem_Alloc(sizeof(*context));
102
103     if (context) {
104         SOS_memset(context, 0, sizeof(*context));
105         context->Remainder = SOS_Message_Create();
106     }
107
108     return context;
109 }
110
111 /*++
112 Routine Name:
113
114     Framer_ContextDestroy
115
116 Routine Description:
117
```

```
119
120 Parameters:
121
122     FRAMER *        Context - [consumed]
123         An Framer context who's resources and memory are to be freed
124
125 Return Value:
126
127     None.
128
129 --*/
130 static
131 void
132 Framer_ContextDestroy(
133     FRAMER *        Context
134 )
135 {
136     SOS_DEBUGOUT_FUNC_TRACE("Framer_ContextDestroy called\n");
137
138     if (Context) {
139         SOS_Message_Destroy(Context->Remainder);
140         SOS_Mem_Free(Context);
141     }
142 }
143
144 /*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
… +++++
145 Edge Handlers
146 ----------------------------------------------------------------------
… ---*/
147
148 /*++
149 Routine Name:
150
151     Framer_KeyCreate
152
153 Routine Description:
154
155     Creates a unique Key and assocates it with the given path every
156     time it is called.
157     A value of SOS_PathBuild_Stop is returned because
158     whoever comes after Framer wants the AudioFormat variable to
159     be set and it currently is not set until Framer_MessageHandler.
160
161 Parameters:
```

```
162
163     SOS_PATH     *Path    - [in]
164         The new Path to create a Key for
165
166     SOS_MESSAGE  *Message - [in]
167         The first message sent along this path.
168
169 Return Value:
170
171     SOS_STATUS -
172         SOS_PathBuild_Stop on success.
173
174         SOS_Error if something goes wrong.
175 --*/
176 static
177 SOS_STATUS
178 Framer_KeyCreate(
179     SOS_PATH     *Path,
180     SOS_MESSAGE  *Message
181 )
182 {
183     static int      s_UniqueId;
184
185     SOS_REGOBJECT*  uniqueSessionKey = SOS_Int32_Create(s_UniqueId++);
186     SOS_STATUS      status           = SOS_Success;
187
188     SOS_DEBUGOUT_FUNC_TRACE("Framer_KeyCreate called\n");
189
190     if (!uniqueSessionKey) {
191         status = SOS_Error;
192     }
193
194     /* set a unique key on the Path */
195     if (SOS_SUCCEEDED(status)){
196         status = SOS_Path_SessionKeySet(Path, uniqueSessionKey);
197         SOS_RegObject_Release(uniqueSessionKey);
198     }
199
200     return status;
201 }
202
203
204 /*++
205 Routine Name:
```

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.