

```
1 /*+++++
... +++++
2
3 $Id: bmptorgb.c,v 1.5 2001/10/02 21:04:10 guyc Exp $
4
5 Copyright (c) 2001 BeComm Corporation
6
7 Filename:
8
9     bmptorgb.c
10
11 Notes:
12
13     currently the bead uses the rgbcontext implementation
14     which does not allow us to convey the bmp-specific
15     palette information. This is okay, since the decoder
16     currently only supports formats that don't use the
17     color info, but to extend this to support all bmp files
18     the context will need to convey the color info
19     somehow - either within the path context or the message.
20
21 Abstract:
22
23     Converts encoded bitmaps to 24-bit RGB bitmaps.
24
25 Owner:
26
27     Guy Carpenter (guyc)
28
29 -----
... ---*/
30
31 #define SOS_DEBUG_ZONE "/beads/bmptorgb"
32 #include <sosstrings.h>
33 #include "bmpformat.h"
34 #include "bmpdecoder.h"
35 #include <sosmultimedia.h>
36
37 SOS_SOURCE_VERSION("$Id: bmptorgb.c,v 1.5 2001/10/02 21:04:10 guyc Exp
... $");
38
39 #define video
40
41 /*+++++
```

```
42 Named Constants
43 -----
44 ... ---*/
45 /*
46  * Name of bead
47  */
48 static const char BEAD_NAME[] = "bmptorgb";
49
50 static const char VIDEOCONTEXT_CLASS_NAME[] = "rgbcontext";
51
52 /*+++++
53  Structs
54 -----
55 ... ---*/
56 typedef struct {
57     BMP_FORMAT           BmpFormat;
58     BMP_DECODER *       BmpDecoder;
59     SOS_IVIDEOCONTEXT * InVideoContext;
60     SOS_IVIDEOCONTEXT * OutVideoContext;
61     SOS_VIDEO_TIMESTAMP FrameNumber;
62 } BMPTORGB_CONTEXT;
63
64
65 /*+++++
66  Context Stuff
67 -----
68 ... ---*/
69 static
70 void
71 BmpToRgb_ContextDestroy(
72     BMPTORGB_CONTEXT * Context
73 )
74 {
75     if (Context) {
76
77         if (Context->BmpDecoder) {
78             BmpDecoder_Destroy(Context->BmpDecoder);
79         }
80
81         SOS_Interface_Release(Context->InVideoContext);
```

```
82     SOS_Interface_Release(Context->OutVideoContext);
83
84     SOS_Mem_Free(Context);
85 }
86 }
87
88 static
89 BMPTORGB_CONTEXT *
90 BmpToRgb_ContextCreate(
91     void
92 )
93 {
94     SOS_STATUS status = SOS_Success;
95     BMPTORGB_CONTEXT *context;
96
97     context = SOS_Mem_Alloc(sizeof(*context));
98     if (context) {
99         SOS_memset(context, 0, sizeof(*context));
100
101         context->BmpDecoder = BmpDecoder_Create();
102         if (!context->BmpDecoder) {
103             /* failed to create decoder */
104             status = SOS_Error;
105         }
106
107         /*
108          * If anything went wrong we should cleanup
109          */
110         if (SOS_FAILED(status)) {
111             BmpToRgb_ContextDestroy(context);
112             context = NULL;
113         }
114     } else {
115         /* failed mallocing context */
116         status = SOS_ErrorResourceAllocation;
117     }
118
119     return context;
120 }
121
122 /*+++++
...
+++++
123
124 +++++
...
+++++

```

```
125 static
126 SOS_STATUS
127 ContextPrepare(
128     BMPTORGB_CONTEXT*    Context,
129     SOS_PATH*            Path
130 )
131 {
132     SOS_STATUS status = SOS_Success;
133     SOS_REGOBJECT *videoContextObject;
134
135     status = SOS_Path_AttributeGet(
136         Path,
137         SOS_VIDEOCONTEXT_NAME,
138         &videoContextObject
139     );
140
141     SOS_ASSERT_SOFT_ERROR(
142         SOS_SUCCEEDED(status),
143         "Path context does not contain a video context"
144     );
145
146     if (SOS_SUCCEEDED(status)) {
147         status = SOS_RegObject_InterfaceGet(
148             videoContextObject,
149             SOS_IVIDEOCONTEXT_ID,
150             (void*)&(Context->InVideoContext)
151         );
152
153         SOS_RegObject_Release(videoContextObject);
154     }
155
156     Context->OutVideoContext = SOS_Interface_CreateFromClassName(
157         VIDEOCONTEXT_CLASS_NAME,
158         SOS_IVIDEOCONTEXT_ID
159     );
160
161
162     SOS_ASSERT_ASSUMPTION(
163         Context->OutVideoContext!=NULL,
164         "Couldn't create video context"
165     );
166
167     if (Context->OutVideoContext) {
168         status = SOS_Path_AttributeSet(
```

```
170         Path,  
171         SOS_VIDEOCONTEXT_NAME,  
172         SOS_Interface_ObjectPeek(Context->OutVideoContext)  
173     );  
174 }  
175  
176     return status;  
177 }  
178  
179  
180 SOS_STATUS  
181 DecodeFrame(  
182     SOS_PATH *          Path,  
183     BMPTORGB_CONTEXT * Context,  
184     SOS_MESSAGE *       Message  
185 )  
186 {  
187     SOS_STATUS status = SOS_Success;  
188     SOS_VIDEO_FORMAT inFormat, outFormat;  
189     SOS_VIDEO_TIMESTAMP timeStamp;  
190  
191     SOS_MESSAGE *outMessage = SOS_Message_Create();  
192     size_t encodedFrameSize = 0;  
193     size_t decodedFrameSize = 0;  
194     void *encodedFrame = NULL;  
195     void *decodedFrame = NULL;  
196  
197     status = Context->InVideoContext->Unpack(  
198         Context->InVideoContext,  
199         Message,  
200         &inFormat,  
201         &timeStamp  
202     );  
203  
204     if (SOS_SUCCEEDED(status)) {  
205         encodedFrameSize = SOS_Message_LengthGet(Message);  
206         decodedFrameSize = inFormat.Width * inFormat.Height *  
207     ... sizeof(SOS_UINT32);  
208  
209         encodedFrame = SOS_Mem_Alloc(encodedFrameSize);  
210         decodedFrame = SOS_Mem_Alloc(decodedFrameSize);  
211     }  
212  
213     if (encodedFrame && decodedFrame) {
```

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.