Subject: A question for coders in the community.
Posted by kerryg on Sun, 17 Jan 2010 20:54:07 GMT
View Forum Message <> Reply to Message

Problems often melt away with enough sunlight on them, so here's a riddle for the community coders.

XN Resource Editor is the application we use to install the skins that Jim Drago, Mike Yanoska and others build for PARIS. A small issue blocks more complete skin work - certain images make PARIS crash if replaced or altered.

Whole sections can be freely altered, but It only takes a trivial alteration in certain areas to provoke a crash. Try this experiment yourself if you're curious:

1) Make a copy of your paris.exe.
2) Download and boot up XN Resource Editor.
3) From inside XN Resource Editor, navigate to the copy of the paris.exe you made and open it.
4) In "PICT" resources you'll find one called MIXER. Take the pencil tool and draw a visible mark in a specific place, shown below (it can be as little as a single pixel, but a bit larger is better for the purpose of visibility of the results).
5) Save the app. Quit XN Resource Editor.
6) Boot PARIS from the altered .exe. Open a Mixer window; you should see your "graffiti" mark displayed above the Auxes. Now - close the mixer window.

PARIS should now quit with ***ASSERTION FAILED: aPalettes.c:317 palette!=NULL.

Questions to those who are code-savvy (I am not, I'm just a layman who isn't afraid to ask stupid questions): why? What changes - palette? Header? File size? DPI (the 8-bit original is an unusual "86 DPI")?

Finding a solution to this will help people skinning PARIS take the work further. Perhaps it's a matter of opening the changed graphic in a hex editor and altering the header/palette to match the original in some way? There are some real talents here in the community; insights gratefully received.

Thanks!

Subject: Re: A question for coders in the community.
Posted by Jim Drago[2] on Sun, 17 Jan 2010 21:34:58 GMT
View Forum Message <> Reply to Message

Kerry, I thought of a new test..

Open XN
Open a COPY of PARIS EXE
Open MIXER BMP
Change nothing and do a Save As, give it the same name and save it to the same place so it simply overwrites itself. Nothing has changed except a saving process.
Start app and test for error.

If the error occurs it has to be because of some renaming in the header info that the newer OS adds or eliminates. It seems to me anyway.

Anyone please chime in and brainstorm with us. Hours have been spent for years trying to work around this.

Another thing worth considering, before 3.0 this BMP provided all of the pieces for the user interface. In 3.0 5 different BMPS provide the artwork. This BMP was very important to the original app, now except for the hashmarks on the knobs and the blanks above the Master and Global sections none of it seems to be used.

Another thought if anyone has a computer running PARIS with Windows 95 or 98 could you run Kerry's test. This might address any renaming the newer OS is doing.

Subject: Re: A question for coders in the community.
Posted by kerryg on Mon, 18 Jan 2010 22:01:13 GMT
View Forum Message <> Reply to Message

Couldn't follow those instructions 100%, but close enough:

- opened the app in XN Resource Editor
- saved it under a different name
- found the graphic
- selected all, cut
- pasted it back in on top of itself
- saved

Sure enough - same error.

So - what have we learned? Maybe something important - maybe that this old problem might not necessarily lie in PARIS after all, but in something XN Resource Editor (or the clipboard itself) does to the 8 bit BMP?

If so, the reason this anomaly showed up when we were editing that particular graphic would be because it's one of the few areas in the app that necessitates editing one of the older 8-bit BMPs that remained in the app at time of discontinuation (the others had already been updated to 24-bit). After a while, folks just learned what to avoid touching - the ADAT, EDITOR, MIXER and ADAT bmps. Folks who have done skinning - does that sound about right?

If that's the case, that'd be a pretty cool realization, since we don't have the source code to PARIS but we do actually have access to the source code of XN Resource Editor; it's right here.

Subject: Re: A question for coders in the community.
Posted by Jim Drago[2] on Mon, 18 Jan 2010 23:35:19 GMT
View Forum Message <> Reply to Message

Correct but I believe the only BMP that triggers the error is the MIXER BMP. Any other original would have to be tested also.

Did you get the article I sent which discusses how BMP headers have changed? If not here it is again

http://en.wikipedia.org/wiki/BMP_file_format

I couldn't follow most of it because I don't understand code speak. But it definately insinuates that BMP header info has changed since earlier OS's.

I think it might be just that simple, that in order to work the BMP's header info has to be edited to match the old requirements. That the app looks for something in the header of that BMP even though for the most part it is not even used anymore. Any one please feel free to correct me as this code stuff is not my thing.

An interesting thing to see is if the MIXER BMPs code info is the same between the original and the original after it has been resaved either through XN or some other program. I don't know if it happens at XN or at the program it is changed in or either. From your above test I think it is just ANY saving in a newer OS.

How do you open the code of a BMP. Sorry but I know nothing about this area.

If the code could be compared between the original BMP and one that had been resaved I'm sure an anomaly between the 2 will appear.

Subject: Re: A question for coders in the community.
Posted by Jim Drago[2] on Tue, 19 Jan 2010 00:01:43 GMT
View Forum Message <> Reply to Message

I found this

general

the bmp file has been created by microsoft and ibm and is therefor very strictly bound to the architecture of the main hardware platform that both companies support: the ibm compatible pc. this means that all values stored in the bmp file are in the intel format, sometimes also called the

little endian format because of the byte order that an intel processor uses internally to store values.

the bmp files are the way, windows stores bit mapped images. the bmp image data is bit packed

zeroes. bmp files are stored bottom-up, that means that the first scan line is the bottom line.

the bmp format has four incarnations, two under windows (new and old) and two under os/2, all are described here.

here

http://atlc.sourceforge.net/bmp.html#_toc381201084

That leads me to believe herein lies the possible problem and solution