PNG WG Meeting Minutes

Aug 8, 2022

Attendees:

(Alphabetical order, black text means attended, red text means missed.)
Chris Blume
Chris Lilley
Chris Seeger
Leonard Rosenthol
Pierre-Anthony Lemieux

Topics & Minutes:

- GitHub issue #150 (Define some amount of error handling behavior)
  - Should we specify error handling at all?
  - How broad should the effects of an error be? Should it be as contained as possible? Should it affect other parts that reference it? Case-by-case?
  - CL: It is good to pull out generalizations in a note that is not a firm requirement but does encourage behavior we want such as minimizing error severity and surface area.
  - PAL: We would perhaps be better making a strong validator. No objection, but this might not provide much benefit since people are already encouraged to do this.
  - We agree there might not be much value in adding wording. Instead, we should add offending issues to a standard test set.
- GitHub issue #96 (Out-of-range palette indices)
  - Most PNG decoders that we know of treat color #100 from a 64-color palette as opaque black. Std_image treats it as transparent. Interestingly, treating it as transparent has become an unofficially recognized behavior of many gif decoders.
  - We previously decided not to overspecify. In this case, a note might be the best solution.
  - This relates to #150 because we could treat the pixel as an error (small scope) or the palette as an error (larger scope).
  - Separately, should we allow trailing unused data in the data stream? Several images in the wild have this. We run into this issue often with zlib, where users write the entire buffer instead of just the used part of the buffer. As a result, many implementations have problems when this unused data isn't zeroed out.
  - PAL: We should be firm on encoders to not do this.

    - We agree that we could add a note mentioning there are images in the wild which use this incorrectly.
    - PAL: The extra-trailing-bytes issue comes up all the time. Sometimes this is actually useful such as in constant bitrate streams. This also comes up in hardware implementations.
- GitHub issue #29 (Ambiguities about extra sample bits in bKGD, tRNS chunks)
    - The spec currently claims that these chunks will always be 16 bit. Even when the image's color depth is less than 16 bit. The spec also says the unused high bits "are 0". But obviously, encoders could write non-zero values here and be non-compliant.
    - Libpng and its users (notably, Chrome), issue a warning but ignore the high bits. Firefox instead treats the whole chunk as having errored and ignores the chunk.
    - CL: Diverging behavior is not ideal. Firefox is perhaps being overly cautious in this case. It shouldn't be an error. And diverging behavior makes PNG look bad in general.
    - We agree that the wording of the spec is bad and we should not claim the bits are zero. We should change it to say something along the lines of "should be zero". It should not be an error case.
- GitHub issue #12 (CRC calculation not well defined)
    - CL proposed in the bug that we mention the code is informative and link to the normative gzip RFC.
    - PAL: I like referencing a definitive source.
- Definition issues
    - Some definitions are completely unused. This is mentioned in GitHub issue #155. Each instance might be a special case so they might each need their own GitHub issue.
    - Some definitions duplicate data. For example a definition with the exact same text as a later section that further expands on the definition. This is mentioned in GitHub issue #156.
    - Some definitions are too vague. This is mentioned in GitHub issue #157.
    - We all agree to keep the definition section but trim it down to only the useful definitions.

Action Items:

- CB will look into how we define the unused bytes at the end of a compression stream and how they can be ignored.
- PAL to look at the problems with definitions.