http://java.sun.com/developer/sampsource/jai/bugs-1_1_1-codec.html Nov 12, 2005

Article

Java Advanced Imaging API Sample Source Known Bugs

Advanced Imaging API Sample Image File Readers and Writers Known Bugs and Issues

These are the known bugs and enhancement requests for the Java Advanced Imaging (JAI) Sample Image File Readers and Writers.

- The JPEG decoder cannot decode abbreviated JPEG streams.
- The FlashPix decoder throws an ArrayIndexOutOfBoundsException when reading some FlashPix images created by Microsoft Picture It 2000.
- The BMP encoder does not support writing out images compliant with any other version other than Version 3.0.
- BMP, TIFF and JPEG encoders cannot encode BufferedImages of the type INT_RGB. A workaround is to re-format the data before passing it to the encoder such that each 8-bit sample is stored in a byte.
- The TIFF encoder does not support LZW compression for the usual reason of the patent on the algorithm.
- TIFF decoder cannot read images stored in Planar format (PlanarConfiguration tag has value 2).

These bugs are present in the JAI 1.1.1 release but have a source level fix provided in the `BugFixes.jar` jar archive contained in the JAI 1.1.1 Sample I/O download bundles.

- The TIFF encoder cannot write 4-bit grayscale images.
- The TIFF encoder does not generate correct G3_2D and G4 compressed data when the image width is not a multiple of 8.
- The TIFF encoder cannot correctly write multiple pages per TIFF file when the data are bilevel and compression is used.
- The TIFF decoder cannot read non-compliant TIFF images wherein the value of the "RowsPerStrip" field incorrectly exceeds the image height.
- The FlashPix decoder throws an index out of bounds exception as a result of ignoring the `ENDOFCHAIN` codes in a FAT sector array.
- The FlashPix decoder does not ignore the color space calibration bit and consequently does not apply color space conversion to uncalibrated images.
- Index out of bounds errors occur in `SimpleRenderedImage` `getData(Rectangle)` and `copyData()` methods. This does not occur when the codecs are used within JAI, i.e., via the I/O operations, e.g., "FileLoad", "FileStore", etc., as in that case these methods are never invoked.