

JChemPaint too: PNG embedded connectivity tables

Egon Willighagen 

Published August 24, 2007

Citation

Willighagen, E. (2007). JChemPaint too: PNG embedded connectivity tables. In *chem-bla-ics*. chem-bla-ics. <https://doi.org/10.59350/rxpr6-t1772>

Keywords

Jchempaint, Cheminf

Abstract

Rich blogged about Firefly embedding MDL molfiles in PNG images , which I found really cool. Rich and Noel later showed how that metadata can be retrieved again , possibly with Python.

Copyright

Copyright © Egon Willighagen 2007. Distributed under the terms of the [Creative Commons Attribution 4.0 International License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

chem-bla-ics

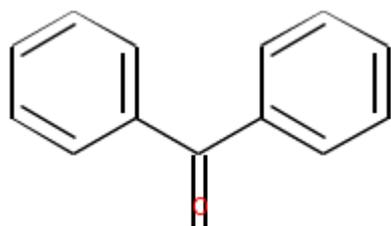
Rich blogged about Firefly [embedding MDL molfiles in PNG images](#), which I found [really](#) cool. Rich and Noel later showed how that metadata [can be retrieved again](#), possibly [with Python](#).

But I did not like that [Firefly](#) could do this, and [JChemPaint](#) not. So, I started hacking. First I discovered I had to get rid of the use of [JAI](#); then I had to adapt the `JChemPaintPanel.takeSnapshot()` API to return a `RendererImage`; and finally, I had to figure out how to write the extra metadata. Now, Firefly is not opensource (yet), so it took me some time to figure out how that was done, and this is how:

```
ImageWriter writer = ImageIO.getImageWriters(
    new ImageTypeSpecifier(awtImage), "png"
).next();
ImageTypeSpecifier specifier = new ImageTypeSpecifier(awtImage);
IIOMetadata meta = writer.getDefaultImageMetadata( specifier, null );

Node node = meta.getAsTree( "javax_imageio_png_1.0" );
IIOMetadataNode tExtNode = new IIOMetadataNode("tEXt");
IIOMetadataNode tExtEntryNode = new IIOMetadataNode("tEXtEntry");
tExtEntryNode.setAttribute( "keyword", "molfile" );
tExtEntryNode.setAttribute( "value", mdlMolfile);
tExtNode.appendChild(tExtEntryNode);
node.appendChild(tExtNode);
meta.mergeTree("javax_imageio_png_1.0", node);
ImageOutputStream ios = ImageIO.createImageOutputStream(
    new FileOutputStream(filename)
);
writer.setOutput(ios);
writer.write( meta, new IIOMetadata(awtImage, null, meta), null );
```

Now I can create my own test files for the [Strigi's ability to extract chemical metadata from PNG images](#). Here is the JChemPaint generator PNG image for [benzophenone](#):



Another issue, unrelated to this patch, is that writing PNG images changes the location of the structure in the JChemPaint editor, and that the placing of the element symbol in image writing is seriously broken. But that will soon be solved with [Niels' new renderer](#).

The metadata looks like:

chem-bla-ics

```
<javax_imageio_png_1.0>
  <IHDR width="208" height="129" bitDepth="8" colorType="RGBA" compress
  interlaceMethod="none"/>
  <tEXt>
    <tEXtEntry keyword="molfile" value=" CDK 8/24/07,15:9 14 15 0 0 0 0 0 0 0999 \
    0 0 247.8231 1020.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 247.8231 1056.0000 0.0000 C
    0 0 0 0 0 0 0 0 0 0 310.1769 1056.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 310.1769 10:
    1074.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 341.3538 1110.0000 0.0000 O 0 0 0 0 0 0
    0 0 0 0 403.7077 1074.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 434.8846 1056.0000 0.00
    C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 403.7077 1002.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 372.5307
    0 2 3 1 0 0 0 0 3 4 2 0 0 0 0 4 5 1 0 0 0 0 5 6 2 0 0 0 0 6 1 1 0 0 0 0 5 7 1 0 0 0 0 7 8 2 0
    2 0 0 0 0 12 13 1 0 0 0 0 13 14 2 0 0 0 0 14 9 1 0 0 0 0 M END "/>
  </tEXt>
</javax_imageio_png_1.0>
```

(Newlines are lost in the XML display.)

JChemPaint does not yet write InChIs, and it also does not open PNG images for input yet (as Firefly does).