chem-bla-ics

CDK-JChemPaint #2: rendering reactions

Egon Willighagen

Published April 5, 2010

Citation
Willighagen, E. (n.d.). In *chem-bla-ics*. chem-bla-ics. https://doi.org/10.59350/wyfzs-jsm57

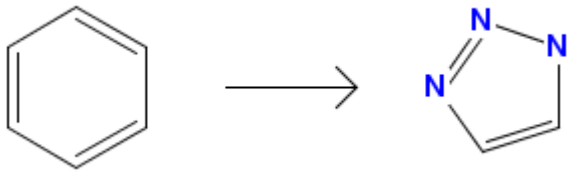
Keywords
Cdk, Jchempaint

Copyright
Copyright © Egon Willighagen 2010. 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

I posted earlier today a Groovy script to render molecules with CDK-JChemPaint 8. Now, the new JChemPaint rendering engine also contains the functionality to render reactions. So, I can also do:

```
$ groovy renderReaction.groovy
```



The matching script:

```
import java.util.List;

import java.awt.*;
import java.awt.image.*;

import javax.imageio.*;
import javax.vecmath.*;

import org.openscience.cdk.*;
import org.openscience.cdk.geometry.*;
import org.openscience.cdk.interfaces.*;
import org.openscience.cdk.layout.*;
import org.openscience.cdk.renderer.*;
import org.openscience.cdk.renderer.font.*;
import org.openscience.cdk.renderer.generators.*;
import org.openscience.cdk.renderer.visitor.*;
import org.openscience.cdk.templates.*;

int WIDTH = 600;
int HEIGHT = 600;

// the draw area and the image should be the same size
Rectangle drawArea = new Rectangle(WIDTH, HEIGHT);
Image image = new BufferedImage(
  WIDTH, HEIGHT, BufferedImage.TYPE_INT_RGB
);

IMolecule benzene = MoleculeFactory.makeBenzene();
IMolecule triazole = MoleculeFactory.make123Triazole();
IReaction reaction = new Reaction();
```

```
StructureDiagramGenerator sdg = new StructureDiagramGenerator();
sdg.setMolecule(triazole);
sdg.generateCoordinates();
triazole = sdg.getMolecule();
sdg.setMolecule(benzene);
sdg.generateCoordinates();
benzene = sdg.getMolecule();
try {
GeometryTools.translate2DCenterTo(benzene, new Point2d(-4,0))
GeometryTools.translate2DCenterTo(triazole, new Point2d(4,0))
} catch (Exception e) {
 e.printStackTrace();
}

reaction.addReactant(benzene);
reaction.addProduct(triazole);

// generators make the image elements
List generators = new ArrayList();
generators.add(new BasicSceneGenerator());
generators.add(new BasicBondGenerator());
generators.add(new BasicAtomGenerator());

 List reactiongenerators =
  new ArrayList();
reactiongenerators.add(new ReactionArrowGenerator());
reactiongenerators.add(new ReactionPlusGenerator());

// the renderer needs to have a toolkit-specific font manager
Renderer renderer = new Renderer(
  generators, reactiongenerators, new AWTFontManager()
);

// the call to 'setup' only needs to be done on the first paint
renderer.setup(reaction, drawArea);

// paint the background
Graphics2D g2 = (Graphics2D)image.getGraphics();
g2.setColor(Color.WHITE);
g2.fillRect(0, 0, WIDTH, HEIGHT);

// the paint method also needs a toolkit-specific renderer
renderer.paintReaction(reaction, new AWTDrawVisitor(g2));
```

```
ImageIO.write(
  (RenderedImage)image, "PNG", new File("reaction.png")
);
```