

Creating CMLReact from UsefulChem Ugi Reactions

Egon Willighagen 

Published August 31, 2008

Citation

Willighagen, E. (2008). Creating CMLReact from UsefulChem Ugi Reactions. In *chem-bla-ics*. chem-bla-ics. <https://doi.org/10.59350/32fcj-fd465>

Keywords

Cml, Cdk, Inchi, Usefulchem

Abstract

Cameron, Jean-Claude and I were invited to Peter's place in Cambridge, where we are now hacking on CMLReact for the Ugi reactions Jean-Claude has been working on. I just finished a script that uses the CDK and Sam's interface to the InChI library to convert a list of four reactants and one Ugi product into CMLReact (doi:10.1021/ci0502698). The full BeanShell script looks like:

Copyright

Copyright © Egon Willighagen 2008. 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

Cameron, [Jean-Claude](#) and I were invited to [Peter's](#) place in Cambridge, where we are now hacking on CMLReact for the [Ugi reactions](#) Jean-Claude has been working on. I just finished a script that uses the CDK and Sam's interface to the [InChI library](#) to convert a list of four reactants and one Ugi product into CMLReact (doi:[10.1021/ci0502698](https://doi.org/10.1021/ci0502698)). The full [BeanShell](#) script looks like:

```
#!/usr/bin/bsh

import java.io.File;
import java.io.FileReader;
import java.io.BufferedReader;

import org.openscience.cdk.*;
import org.openscience.cdk.exception.*;
import org.openscience.cdk.inchi.*;
import org.openscience.cdk.interfaces.*;
import org.openscience.cdk.io.CMLWriter;

import org.openscience.cdk.libio.cml.Convertor;
import org.xmlcml.cml.element.CMLReaction;

import net.sf.jniinchi.INCHI_RET;

InChIGeneratorFactory factory = new InChIGeneratorFactory();
// Get InChIToStructure

File file = new File("inchi.ugi.txt"); // five inchis expected, last being the product
BufferedReader reader = new BufferedReader(new FileReader(file));

String first = reader.readLine();
String second = reader.readLine();
String third = reader.readLine();
String fourth = reader.readLine();
String product = reader.readLine();

System.out.println("First: " + first);
IMolecule firstAC;
{
    InChIToStructure intostruct = factory.getInChIToStructure(first, DefaultChemObjectBuilder

    INCHI_RET ret = intostruct.getReturnStatus();
    if (ret == INCHI_RET.WARNING) {
        // Structure generated, but with warning message
```

chem-bla-ics

```
        System.out.println("InChI warning: " + intostruct.getMessage());
    } else if (ret != INCHI_RET.OKAY) {
        // Structure generation failed
        throw new CDKException("Structure generation failed failed: " + ret.toString()
            + " [" + intostruct.getMessage() + "]");
    }

    firstAC = new Molecule(intostruct.getAtomContainer());
}

System.out.println("Second: " + second);
IMolecule secondAC;
{
    InChIToStructure intostruct = factory.getInChIToStructure(second, DefaultChemObjectBuilder

    INCHI_RET ret = intostruct.getReturnStatus();
    if (ret == INCHI_RET.WARNING) {
        // Structure generated, but with warning message
        System.out.println("InChI warning: " + intostruct.getMessage());
    } else if (ret != INCHI_RET.OKAY) {
        // Structure generation failed
        throw new CDKException("Structure generation failed failed: " + ret.toString()
            + " [" + intostruct.getMessage() + "]");
    }

    secondAC = new Molecule(intostruct.getAtomContainer());
}

System.out.println("Third: " + third);
IMolecule thirdAC;
{
    InChIToStructure intostruct = factory.getInChIToStructure(third, DefaultChemObjectBuilder

    INCHI_RET ret = intostruct.getReturnStatus();
    if (ret == INCHI_RET.WARNING) {
        // Structure generated, but with warning message
        System.out.println("InChI warning: " + intostruct.getMessage());
    } else if (ret != INCHI_RET.OKAY) {
        // Structure generation failed
        throw new CDKException("Structure generation failed failed: " + ret.toString()
            + " [" + intostruct.getMessage() + "]");
    }

    thirdAC = new Molecule(intostruct.getAtomContainer());
```

chem-bla-ics

```
}

System.out.println("Fourth: " + fourth);
IMolecule fourthAC;
{
    InChIToStructure intostruct = factory.getInChIToStructure(fourth, DefaultChemObjectBuilder.getInstance());

    INCHI_RET ret = intostruct.getReturnStatus();
    if (ret == INCHI_RET.WARNING) {
        // Structure generated, but with warning message
        System.out.println("InChI warning: " + intostruct.getMessage());
    } else if (ret != INCHI_RET.OKAY) {
        // Structure generation failed
        throw new CDKException("Structure generation failed failed: " + ret.toString()
            + " [" + intostruct.getMessage() + "]);
    }

    fourthAC = new Molecule(intostruct.getAtomContainer());
}

System.out.println("Product: " + product);
IMolecule productAC;
{
    InChIToStructure intostruct = factory.getInChIToStructure(product, DefaultChemObjectBuilder.getInstance());

    INCHI_RET ret = intostruct.getReturnStatus();
    if (ret == INCHI_RET.WARNING) {
        // Structure generated, but with warning message
        System.out.println("InChI warning: " + intostruct.getMessage());
    } else if (ret != INCHI_RET.OKAY) {
        // Structure generation failed
        throw new CDKException("Structure generation failed failed: " + ret.toString()
            + " [" + intostruct.getMessage() + "]);
    }

    productAC = new Molecule(intostruct.getAtomContainer());
}

IReaction ugiReaction = new Reaction();
ugiReaction.addReactant(firstAC);
ugiReaction.addReactant(secondAC);
ugiReaction.addReactant(thirdAC);
ugiReaction.addReactant(fourthAC);
ugiReaction.addProduct(productAC);
```

chem-bla-ics

```
StringWriter stringWriter = new StringWriter();  
CMLWriter cmlWriter = new CMLWriter(stringWriter);
```

```
cmlWriter.write(ugiReaction);  
cmlWriter.close();  
System.out.println(stringWriter.toString());
```

My apologies for the code duplication, but never tried inline functions in BeanShell yet... You can monitor the efforts at [Google Docs](#).