chem-bla-ics

Molecular Inorganics: SMILES, MDL molfile v3000, and InChIs

Egon Willighagen

Published July 26, 2026

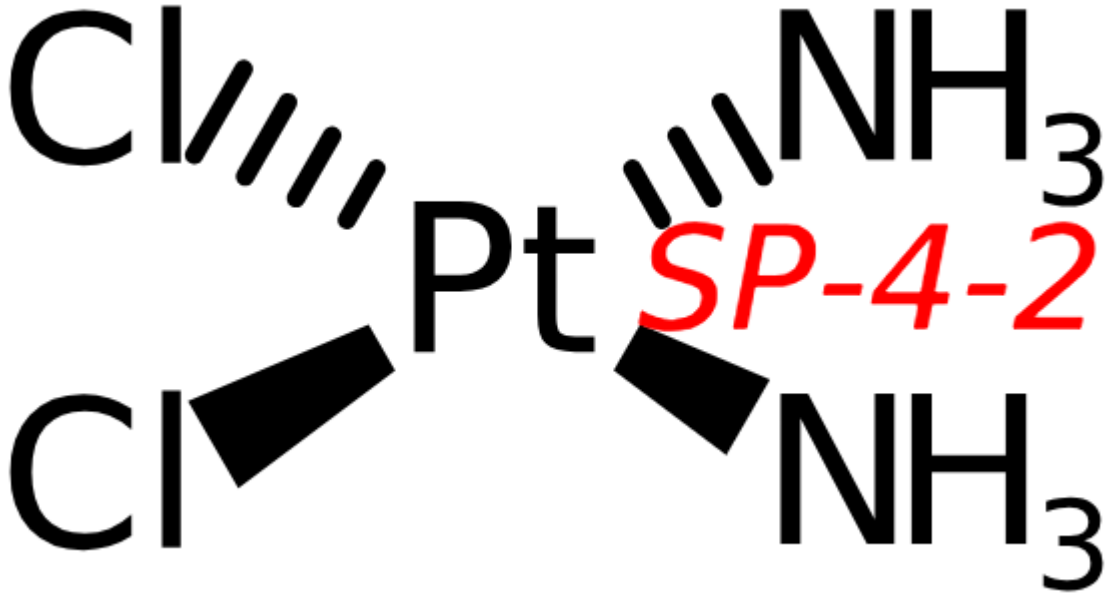
Citation
Willighagen, E. (n.d.). In *chem-bla-ics*. chem-bla-ics. https://doi.org/10.59350/yhf27-fp921

Keywords
Chemistry, Inchi, Wikidata, Smiles, Pubchem

Making chemistry more FAIR requires unique identifiers for chemical structures. For organic compounds plenty of solutions exist that do a great job. Last year and last week, I attended two technical InChI meetings, both with organometallic compounds as one of the key topics. Thanks to Sonja (Mastodon bridge) and Gerd for the invitations. My role includes thinking about what all the work on the InChI means for the Chemistry Development Kit.

Many things came up. One was testing of new InChI functionality for these organometallic compounds, particularly the stereochemistry. Wikipedia has many chemical compounds and could be a source, but so does Wikidata. Both use the SMILES, but not all SMILES captures all the chemistry we need. And the InChI software needs an V3000 MDL Molfile. Thanks to John and other CDK developers, there is good support for recent cheminformatics software, but I was not sure it had what I would need.

This post is the first of a few related posts. This post is about converting SMILES from Wikidata to V3000 files. Take cisplatin: it has four ligands around a platinum atom, all in a single plane:



In this image, in red, is actually an annotation of how the ligands are oriented around the platinum. This is also reflected in the *isomeric SMILES* in Wikidata: `Cl[Pt@SP1]([NH3])([NH3])Cl`.

The following source code is written in Groovy which I have used for many years because it is less verbose than Java. First, we set up our helper classes:

```
@Grab(group='org.openscience.cdk', module='cdk-smiles', version='2.12')
@Grab(group='org.openscience.cdk', module='cdk-silent', version='2.12')
@Grab(group='org.openscience.cdk', module='cdk-ctab', version='2.12')
@Grab(group='org.openscience.cdk', module='cdk-sdg', version='2.12')

import org.openscience.cdk.smiles.SmilesParser;
```

chem-bla-ics

```
import org.openscience.cdk.silent.SilentChemObjectBuilder;
import org.openscience.cdk.io.SDFWriter;
import org.openscience.cdk.layout.StructureDiagramGenerator;
import javax.vecmath.Vector2d

builder = SilentChemObjectBuilder.getInstance()
sp = new SmilesParser(builder)
sdg = new StructureDiagramGenerator();
```

With some extra code, I can actually get many compounds from Wikidata to convert to v3000 with a SAPRQL (as I have done last year with CXSMILES and polymers, unpublished), but let's go with a single example:

```
smiles = "Cl[Pt@SP1]([NH3])([NH3])Cl"
label = "cisplatin"
wdItem = "Q412415"
```

I can parse the SMILES and generated 2D coordinates with (which is also the approach by CDK Depict which I used for the above 2D diagram of cisplatin):

```
mol = sp.parseSmiles(smiles)
sdg.setMolecule(mol);
sdg.generateCoordinates(new Vector2d(0, 1));
mol = sdg.getMolecule();
```

If you have more than one molfile, they can be combined into a SD file, to which additional properties can be added:

```
mol.setTitle(label)
mol.setProperty("PUBCHEM_SUBSTANCE_SYNONYM", label)
mol.setProperty("PUBCHEM_SUBSTANCE_COMMENT", smiles)
mol.setProperty("PUBCHEM_EXT_DATASOURCE_REGID", wdItem)
mol.setProperty("PUBCHEM_EXT_SUBSTANCE_URL", "https://qlever.scholia.wiki/" + wdItem)
```

And then generate the actual SD file with:

```
writer = new FileWriter(new File("demo.sdf"))
SDFWriter sdfWriter = new SDFWriter(writer);
sdfWriter.getSetting(SDFWriter.OptAlwaysV3000).setSetting("true");
sdfWriter.write(mol);
sdfWriter.close();
writer.close();
```

We then get this v3000 file:

```
cisplatin
  CDK     07262618042D
```

```
  0  0  0     0  0            999 V3000
M  V30 BEGIN CTAB
M  V30 COUNTS 5 4 0 0 0
M  V30 BEGIN ATOM
M  V30 1 Cl -1.29904 2.25 0 0
M  V30 2 Pt 0 1.5 0 0
M  V30 3 N 1.29904 2.25 0 0 VAL=4
M  V30 4 N 1.29904 0.75 0 0 VAL=4
M  V30 5 Cl -1.29904 0.75 0 0
M  V30 END ATOM
M  V30 BEGIN BOND
M  V30 1 1 2 1 CFG=3
M  V30 2 1 2 3 CFG=3
M  V30 3 1 2 4 CFG=1
M  V30 4 1 2 5 CFG=1
M  V30 END BOND
M  V30 END CTAB
M  END
> <PUBCHEM_SUBSTANCE_COMMENT>
Cl[Pt@SP1]([NH3])([NH3])Cl

> <PUBCHEM_EXT_DATASOURCE_REGID>
Q412415

> <PUBCHEM_SUBSTANCE_SYNONYM>
cisplatin

> <PUBCHEM_EXT_SUBSTANCE_URL>
https://qlever.scholia.wiki/Q412415

$$$$
```

I can copy/paste the resulting v3000 content to the InChI Web Demo to calculate the Standard InChI:

```
InChI=1S/2ClH.2H3N.Pt/h2*1H;2*1H3;/q;;;;+2/p-2
```

And this is what the two technical meetings I attended were about: *molecular inorganics*. The above InChI does not feel right, and certainly lost the connectivity of the ligands with the platinum. However, if we add the beta option `-MolecularInorganics`, then we get this InChI (where the `B` in `InChI=1B` reflects the beta state of this feature):

```
InChI=1B/Cl2H6N2Pt/c1-5(2,3)4/h3-4H3
```

However, this beta version does not distinguish cisplatin from transplatin. For that, we need to dive into how to represent the stereochemistry of these inorganics first.