

The CDK data classes and change notifications

Egon Willighagen 

Published April 12, 2006

Citation

Willighagen, E. (2006). The CDK data classes and change notifications. In *chem-bla-ics*. chem-bla-ics. <https://doi.org/10.59350/3xsm0-bt478>

Keywords

Cdk, Cheminf, Jchempaint

Abstract

The data classes of the Chemistry Development Kit are mutable, unlike those of Octet. This means that other classes may need to respond when the content updates. For example, a render class. CDK's ChemObject provides a notifyChanged() and addListener() methods for this.

Copyright

Copyright © Egon Willighagen 2006. 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

The data classes of the [Chemistry Development Kit](#) are mutable, unlike those of [Octet](#). This means that other classes may need to respond when the content updates. For example, a render class. CDK's [ChemObject](#) provides a `notifyChanged()` and `addListener()` methods for this. However, as was [recently](#) pointed out, while this is useful in editors, such as [JChemPaint](#), this is a performance killer in high-throughput situations, such as descriptor calculation, or structure diagram generation runs.

To address this, the [IChemObject](#) interface has been extended with the methods `setNotification(boolean)` and `getNotification()`, which allow to temporarily disable change notifications. There are no helper methods yet to disable it for a complete data structure, like `ChemModelManipulator.setNotification(ChemModel, boolean)`, but I expect these to be written soon.

Alternatively, special data classes may be used if notification is never needed for a special setup, for example, in case the QSAR descriptor calculation. In such cases, the new [NoNotificationChemObjectBuilder](#) can be used:

```
IChemObjectReader reader = new MDLReader(new FileInputStream(new File("some.mol")));
IChemObjectBuilder builder = NoNotificationChemObjectBuilder.getInstance();
IMolecule molecule = reader.read(builder.newMolecule());
// then perform some operation in which the molecule changes a lot
```

The advantage is that you do not have to manually disable notification for each class you instantiate. This should give a considerable speed up, and I hope soon to give some statistics.