

CDK 1.2 to 1.4 API changes #1: creating objects with an IChemObjectBuilder

Egon Willighagen 

Published October 27, 2010

Citation

Willighagen, E. (n.d.). In *chem-bla-ics*. chem-bla-ics. <https://doi.org/10.59350/rvgrt-8b910>

Keywords

Cdk, Cheminf, Java

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.

Later this year (planned) a new stable branch of the CDK library will be released. Time to look at some API changes, to ease migration. In this first post of the series, I will show how the `IChemObjectBuilder` functionality has changed.

CDK 1.2 code

```
IChemObjectBuilder builder =  
    DefaultChemObjectBuilder.getInstance();  
IMolecule molecule = builder.newMolecule();  
molecule.addAtom(builder.newAtom("C"));
```

CDK 1.4 code

```
IChemObjectBuilder builder =  
    DefaultChemObjectBuilder.getInstance();  
IMolecule molecule = builder.newInstance(  
    IMolecule.class  
);  
molecule.addAtom(  
    builder.newInstance(IAtom.class, "C")  
);
```

Now, please note that the `builder.newInstance()` method may actually return null. This is not the case for the [DefaultChemObjectBuilder](#), or the [NoNotifiationChemObjectBuilder](#), but future releases may have dedicated builders that do have such functionality. However, these builder would not supposed to be used for building molecules anyway.

The general patterns of `newInstance()` calls is that the first argument is the interface for which you want an instance. All further parameters are passed as parameters for the object's constructor. The builder maps the input to appropriate class constructors. To know what parameters you can pass when instantiating an `IAtom` with the `DefaultChemObjectBuilder`, you would look at the constructor of [Atom](#). Therefore, we can also call:

```
IAtom atom = builder.newInstance(  
    IAtom.class, "C", new Point2d(0,0)  
);
```