

# A JChemPaint Hack-a-thon

Egon Willighagen 

Published September 2, 2007

## Citation

Willighagen, E. (2007, September 2). A JChemPaint Hack-a-thon. *Chem-bla-ics*. <https://doi.org/10.59350/xa92q-jv085>

## Keywords

Jchempaint

## Abstract

Niels and I held a JChemPaint hack-a-thon today (the IRC log). We had a quite ambitious agenda:

## Copyright

Copyright © Egon Willighagen 2007. 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.

Niels and I held a JChemPaint [hack-a-thon](#) today ([the IRC log](#)). We had a quite ambitious agenda:

- make the renderer modular
- make the controller modular
- make a controller interface with Swing + SWT implementations

All this to make the [JChemPaint](#) editor module of the [CDK](#) more easily integrate with non-Swing widget environments. We achieved to make about 50% of these goals: the controller is now modular, and the [Controller2DHub](#) (soon going to deprecate the old [Controller2D](#)) no longer receives Swing mouse events, but local events by implementing the new [IMouseEventRelay](#) interface.

Controller modules implement the new [IController2DModule](#) interface. This modularization allows a clean up of CDK source code, making it more readable and easier to maintain. This was attempted in the past by setting up an [AbstractController2D](#) and a [SimpleController2D](#). The new approach, however, allows to make separate modules for each rendering mode, which are independent anyway. The old code still needs to be ported to the new architecture, and this is expected to happen in the next two weeks.

Another clean up in the architecture is that the controller modules no longer directly act on the [IChemModel](#), but use a new (badly named) [IChemModelRelay](#) interface, making the architecture more closely adhere to the [MVC concept](#). The [IChemModelRelay](#) API currently contains only two methods, but this is expected to expand considerably, because all current JChemPaint edit actions will have to be passed via this interface.

If you want to give the new architecture a test run, look for the [TestEditor](#) application. At the time of writing, it uses a demo module, the [DumpClosestObjectToSTDOUTModule](#), which dumps the nearest [IAtom](#) to STDOUT.