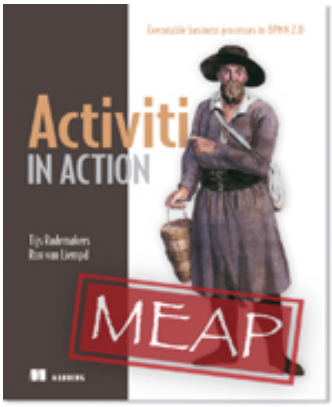
MANNING PUBLICATIONS



Activiti in Action
By Tijs Rademakers and Ron van Liempd

You want to be able to change the content of a rule without too much hassle so you can have direct influence on the behavior of your business processes. In this article, based on chapter 12 of Activiti in Action, the authors show you how Activiti and Drools integrate nicely together to achieve this flexibility with just a few lines of code.

Creating a Web-Based Rule Editor

You want to be able to change the content of a rule without too much hassle so you can have direct influence on the behavior of your business processes.

In this article, we will introduce a small application that enables you to edit the rules deployed on the Activiti engine from a web application and directly deploy the changed rule on the Activiti Engine. No hassle at all! Activiti and Drools integrate nicely together to achieve this flexibility with just a few lines of code.

Introducing flexibility with a custom rule authoring application

In the source code available here, you can find a small Vaadin web application in the `book-rules-app` project that you can run on a jetty web container with Maven. Vaadin is a web framework that has a server-driven programming model that enables you to create fancy web applications with just Java code. To get started with Vaadin, you can download the book of Vaadin at `http://vaadin.com/book`.

The idea of the application is to have a view on the installed Drools rule files and have a way of opening them in the web application, edit the rules and deploy the edited rules right away. This way, no coding in an IDE is necessary; rules can be directly viewed and deployed to speed up the process of change.

You can start the application from the command line with Maven using the `mvn clean install jetty:run` command. Maven will start compiling the code, package the WAR file, start a Jetty server instance, and deploy the WAR file to it.

To really learn what is going on in the Vaadin application, the web book we just mentioned is a great start, but what is interesting to know now is that Vaadin applications have a base class that extends from the Vaadin `Application` class. That class is declared in the `web.xml` and forms the starting point of a Vaadin application, and from there the user interface starts to build up.

Make sure your Activiti Engine is still up and running with the loan request example deployed when you start the rules web application and go to the `http://localhost:8081/book-rules-app` URL. Click the `Drools Rules` button and you will see that on the right sight of the screen a new panel is loaded. In the panel, a table that contains the installed BAR files on the Activiti instance running on Tomcat is populated. In this case, we are interested in an installed loan request process example. Take a look at figure 1 to see where we are.

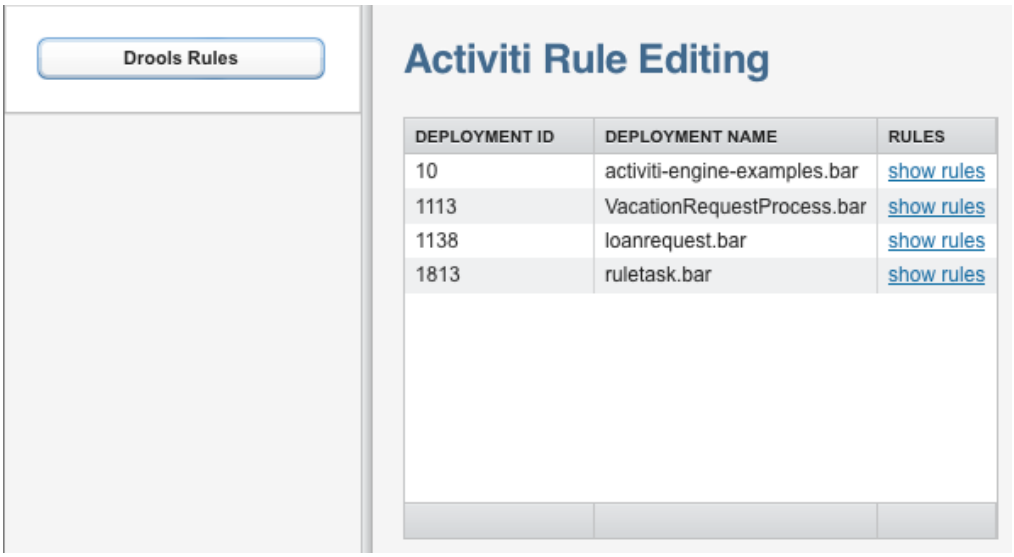


Figure 1 Showing Activiti deployments on the Activiti rule editing panel with help of Vaadin

You can click the **show rules** link in the table. It looks for `.drl` files in the installed archives and will either display a message saying that there are no rules installed or display a second table containing the names of the Drools rule files. When you select the `ruletask.bar` file, you can select a Drools file by clicking the **show rules** link and you will find yourself looking at the screen in figure 2.

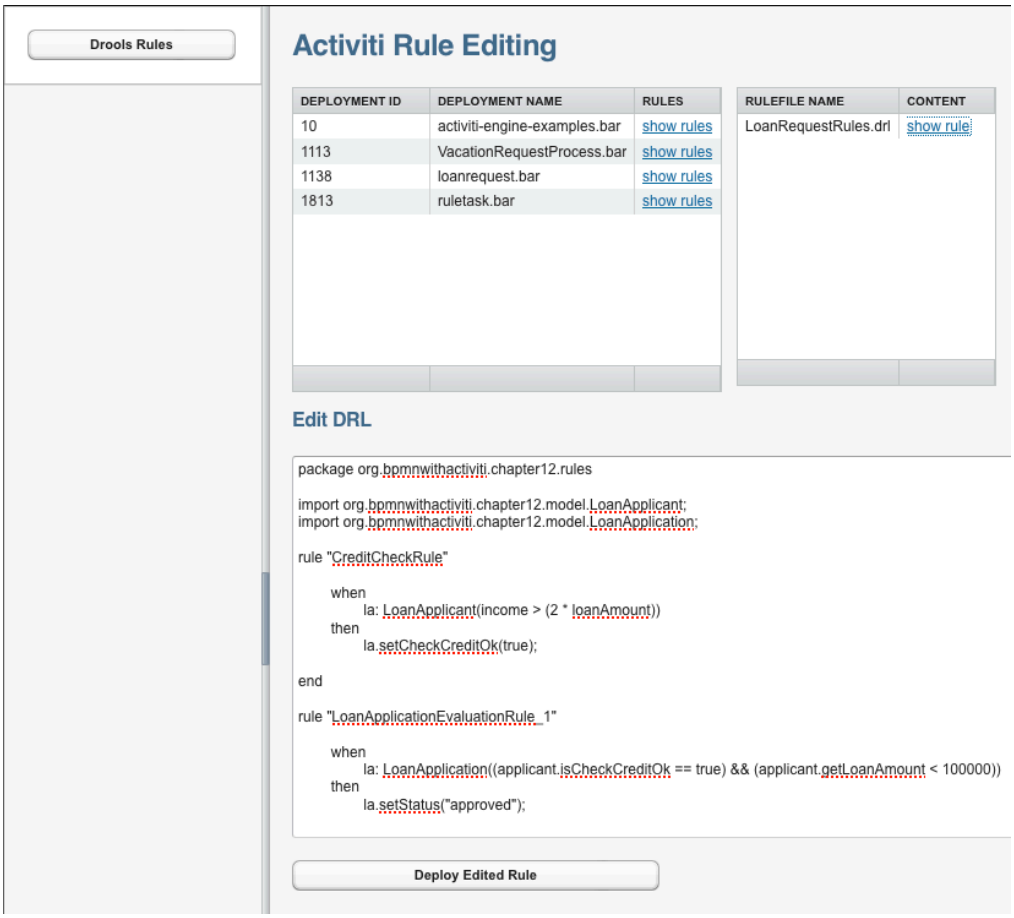


Figure 2 Web-based rule authoring and direct deployment of the edited rules on Activiti Engine

In the text field below `Edit DRL`, you can directly edit the rule now. You can, for example, change the `CreditCheckRule` by making it easier for applicants to get a loan by stating that the applicant's income should be higher than three times the requested amount. After you are done, you can click the `Deploy Edited Rule` button; the edited rule is deployed to the Activiti Engine! You can check it out by running another instance of the loan request process and you will see that the rule is effective immediately! Using a simple application like this simplifies the process of changing the business rules quite a bit.

Summary

While a business rule may be informal or even unwritten and exist only in the heads of people, applying those rules and getting that rule knowledge clear and managed is very valuable. We implemented a simple web application to change deployed rules on the Activiti Engine on the fly.

Here are some other Manning titles you might be interested in:

Open-Source ESBs in Action
Tijs Rademakers and Jos Dirksen

Open Source SOA
Jeff Davis

Mule in Action
David Dossot and John D'Emic

Last updated: November 28, 2011