Validator14ProjectPlan

(From the commons-dev list: Niall Pemberton's thoughts on what should be validator 1.4🙂

The two main goals I had for the next validator release was:

1) remove the dependency on ORO for regex support by moving to a minimum dependency of JDK 1.4 and using java's built in regex.

2) Refactor remaining validation routines into the "o.a.c.routines" package and deprecate the older ones in o.a.c package.

There are three remaining validation routines which need to be refactored into the new package: Credit Card, Email and URL. As part of the refactoring into the new package I took the opportunity to re-write/improve the validation routines that I've done so far and was hoping to do the same with the remaining three.

These three IMO should be broken out into smaller validation routines. For example both URL and Email validation includes validating an IP address and that logic is useful in its own right and should be factored out. The same goes for the check digit validation which is part of the credit card validator. I made a start on this process by factoring out check digit validation[1] and creating a generic [CodeValidator][2] (which combines regex, min/max length and check digit). I also have some stuff in-progress that I never committed (e.g. an IPv4 validator).

[1] http://tinyurl.com/yqdhg8
[2] http://tinyurl.com/25zo2u

This is the vague plan that I had for the remaining routines:

1) Beak out IP address and hostname validation into their own routines. 2) Refactor Emal and URL validation to use the same IP address/hostname validation 3) Refactor the credit card validator to use the new check digit validation OR perhaps to use the new CodeValidator

The one issue that I haven't looked at or worked out what to do about is the logic in the Email validator which strips out comments (see stripComments() method) - I'm not even sure that logic works correctly and it also uses an ORO "substitue" method iteratively.

Lastly once the above is done then I was planning on switching the old validations to use the new versions in the routines package - and deprecate them. Also decide on a plan of what to do with the GenericValidator and GenericTypeValidator - we could leave them unchanged or provide something equivalent in the new routines package

- I had a vague idea to combine them into one class with the methods from GenericValidator prefixed with "is" (they return boolean) and the GenericTypeValidator methods prefixed with "validate" (return an object) - which would be consistent with what I've done in other parts of the new "routines" package.