

WoodyNotes

Various Woody notes

Various nice-to-knows that don't fit somewhere else yet.

Referencing other fields in expressions

Many of Woody's validation rules support using expressions. For example, the "registration form" example uses this to check that the two provided passwords are the same:

```
<wd:datatype base="string">
  <wd:validation>
    <wd:assert test="password = confirmPassword">
      <wd:failmessage>The two passwords are not equal.</wd:failmessage>
    </wd:assert>
  </wd:validation>
</wd:datatype>
```

In the expression, "password" and "confirmPassword" are variables that refer to other widgets. There is a potential problem when one or both these fields don't have a value yet, thus if they are "null". If the referenced widgets are required, then Woody is smart enough to wait with the evaluation of the validation rule until all these widgets have a value. So in that case, there are no problems. If on the other hand the widgets are not required, and the widgets have no value yet, the variables will be replaced with null. If the expression doesn't check on this explicitly, the evaluation of the expression could fail (e.g. with a NullPointerException). [TODO: we need to provide a way to check in expressions if something is null – for now its best to only refer to required widgets].

Expression syntax

The assert expression is evaluated by the `org.outerj.expression` package originally developed for xReporter. It uses an Excel-like syntax <http://xreporter.cocoondev.org/en/expressions.html> where you need to know that:

- Function names are capitalized and case-sensitive
- Boolean operations are not operators but functions (And, Or, Not)
- String constants only allow "value", not 'value'

```
<wd:assert test='
  Or(radiobutton = "value1"
    , And(radiobutton = "value2", Length(otherfield) > 2))' />
```

Form instances

This is an implementation detail, but may be interesting to know: internally each widget in Woody has a "definition" and an "instance". The definition holds all configuration details for a specific widget on a certain form. The instance holds the data for one specific use (instance) of the form. The definition is shared by all instances. This makes the instances very light since they only contain what's absolutely necessary. This distinction between "definition" and "instance" is in fact very similar to Java's classes and objects.