

ValidateThis 0.98 - A Ton of New Stuff, and Two Important Changes

Posted At : November 22, 2010 4:37 PM | Posted By : Bob Silverberg

Related Categories: ColdFusion, ValidateThis

I've just released version 0.98 of [ValidateThis](#), my validation framework ColdFusion. This update has been long in the making, and has been contributed to significantly by a number of folks including [Adam Drew](#), [Marc Esher](#), [Chris Blackwell](#) and [John Wish](#). As always, the latest version can be downloaded from [the ValidateThis RIAForge site](#). There are so many new features and enhancements that I'm going to divide them up into a number of posts, so this post will just include a summary, as well as a discussion of a couple of important changes that will impact all existing users. I'm going to start with a discussion of these changes.

The Folder Structure of the Distribution has Changed

A big thanks to Adam Drew for reorganizing the code repository and the distribution for the framework. This change won't impact any of your code, but it does mean that what you do with the download, in terms of which folders go where, has changed. The framework itself is now contained in the root of the distribution, whereas before it was in a folder called ValidateThis. This means that you can simply unzip the entire zip file into your local /ValidateThis folder. The sample applications are now contained in a folder called /samples, so if you don't want to include these in your project, simply delete this /samples folder after unzipping, or move it elsewhere.

A Change to Metadata that Needs Your Attention

In order to add a cool new enhancement (dynamic parameters) I've had to revisit the way that metadata for parameters was being specified. This examination led me to change the format of the *param* element. Because of this format change, any rules that you may currently have defined that use parameters will need to be changed as well, in order to use this and all future releases of the framework.

The good news is that the change is simple and easy, and Marc Esher has even contributed a regular expression that can be used to automatically update all of your xml files via Eclipse. The truth is that the previous format was actually flawed, so not only will this new format enable this cool new feature, but it will also position the framework better for future enhancement.

The specific change is as follows: It is now a requirement to specify a *name* and *value* attribute for the *param* element, whereas prior to this version neither were required (nor even supported).

To illustrate this change, the following examples of rules using *param* elements in both the old format and the new format, both in xml and json, have been provided:

Old (unsupported) xml format:

```
<property name="UserPass" desc="Password">
  <rule type="rangelength">
    <param minlength="5" />
    <param maxlength="10" />
  </rule>
</property>
```

New xml format:

```
<property name="UserPass" desc="Password">
  <rule type="rangelength">
    <param name="minlength" value="5" />
    <param name="maxlength" value="10" />
  </rule>
</property>
```

Old (unsupported) json format:

```
{ "name": "UserPass", "desc": "Password",
  "rules" : [
    { "type": "rangelength",
      "params" : [
        { "minlength": "5" },
        { "maxlength": "10" }
      ]
    }
  ]
}
```

New json format:

```
{ "name": "UserPass", "desc": "Password",
  "rules" : [
```

```
{ "type": "rangelength",
  "params" : [
    { "name": "minlength", "value": "5" },
    { "name": "maxlength", "value": "10" }
  ]
}
```

What You Need To Do

So, what do you have to change? Based on the format change described above, you'll have to change any rules in your files that use the *param* element. This is a fairly simple change, and, as I mentioned, Marc Esher has provided a regular expression that can be used with Eclipse's find and replace feature. In the Eclipse dialog, use the following values:

Find:

```
<param (.*)="(.*)"
```

Replace:

```
<param name="$1" value="$2"
```

Make sure the "Regular Expressions" checkbox is checked.

Summary of Enhancements

With that out of the way, here is a quick summary of the enhancements and changes included in this release:

- New defaultFailureMessagePrefix configuration option allows you to change the prefix on all failure messages generated by the framework.
- Dynamic parameters for rules are now supported. Parameters can be specified as values, expressions or property names.
- Methods used with the Custom validation type can now return simply true or false.
- Some methods have been added to the framework facade to allow for debugging, testing, administration and extension.
- Removed the jQuery.noConflict() code as it was causing issues in some installations.
- Added a getRequiredPropertiesAndDescs(context) method to the BOValidator.
- Updated the Range validation type to allow for date ranges.
- Changes have been made to the structure of client-side validators (ClientRuleScripters) - custom built CRSs may need to be updated.
- Added a hasParameter(name) and an addParameter(name,value,type) method to the Validation object, to be used by custom ServerRuleValidators.
- Fixed a bug fix in rules/context ordering.
- Added a loadValidators() method to the ValidationFactory object, to allow for pre-loading of validators.
- Reorganized the repo and distribution changing the location of the framework, the sample apps and the tests.
- The VT facade is now available inside the Validation object - can be used for development of new validation types.
- Updated the EqualTo validation to pick up the ComparePropertyDesc from the validation metadata if it is not explicitly specified.
- Changed the Numeric validation type to use isNumeric() instead of isValid("Numeric"), and changed the Date validation type to use isDate() instead of isValid("Date").
- Added a createWrapper(struct|object) function to the ValidationFactory and exposed it through the facade to allow a developer to prepare objects|structures for validation using a BOValidator directly.
- When validating a struct you can now pass in a JSON string the contains a struct in addition to a native CF struct.
- New validation types have been added. These are described in the [VT documentation](#), and will also be described in greater detail in a future post.
 - DateRange
 - DoesNotContainOtherProperties
 - False
 - FutureDate
 - InList
 - IsValidObject
 - NoHTML
 - NotInList
 - PastDate
 - Patterns
 - Size
 - True
 - URL
 - Expression

Once again, the latest code is available from [the ValidateThis RIAForge site](#), and if you have any questions about the framework, or suggestions for enhancements, please send them to the [ValidateThis Google Group](#). And I'd like to again thank Adam, Marc, Chris and John for their contributions to the framework.