ValidatorVersion120

This page describes Changes in version 1.2.0 of Commons Validator (see section **1. Changes**) and notes on Upgrading from Validator 1.1.4 (see section **2 . Upgrading**).

---

1. Changes

N.B. Validator version *1.3.0* has been released with a number of bug fixes for version 1.2.0 and it is recommended that users upgrade to version **1.3.0**

The aim of this section is to provide a summary of the changes in **Validator 1.2.0** - for a full list of the changes in this version see the Changes Report.

1.1 Validator Inheritance

Probably the most significant change in Validator 1.2.0 is the introduction of *inheritance* which comes in two forms:

* **Explicit** Form inheritance
* **Implicit** *Locale* inheritance

1.1.1 Explicit Form Inheritance

Explicit Form Inheritance is where one Form can inherit from another by specifying the name of the form to inherit using the new **extends** property (which is also an new attribute in the validator_1_2_0.dtd). See #27870.

In the following example **nameForm** inherits from **baseForm**, overriding the **lastName** field...

```
<formset>
    <form name="baseForm">
        <field property="firstName" depends="required">
            ...
        </field>
        <field property="lastName" depends="required">
            ...
        </field>
    </form>

    <form name="nameForm" extends="baseForm">
        <field property="lastName" depends="required,minLength">
            ...
        </field>
    </form>
</formset>
```

1.1.2 Implicit Locale Inheritance

Prior to Validator 1.2.0, forms defined in different **FormSet**'s (i.e. different *Locales*) were treated completely separtely. Now Forms with the same name *impl icitly* inherit from those in *less specific* locales.

For example, if you have a form named CustomerForm defined in a FormSet for the **French** language (no country) and also in a FormSet for the **Candian French** locale then the *Candian French* version of the Form will inherit from the *French* one.

See Bug 30955, Bug 16920 and Bug 16920.

N.B. See Section **2. Variables in Resource Bundles** which may remove your need to have multiple FormSet's for different locales altogether.

1.2 Variables in Resource Bundles

One motivation for having different Form Sets was due to the fact that **variable** values needed to be different for different locales. Validator 1.2.0 has the facility to specify variable values in resource bundles which removes the need for multiple formsets in this instance

Two new properties have been added to Variable - **resource** and **bundle**. Setting **resource** to **true** (default is *false*) indicates that the variable's value is a *r esource key*. Additionally the (optional) **bundle** property can be set to the name of an alternative resource bundle.

In the following example the value of **datePattern** variable is specified in the **validatorVariables** bundle under a key of **dateOfBirth.pattern**.

```
    <formset>
        <form name="EmployeeForm">
            <field property="dateOfBirth" depends="date">

                <var resource="true" bundle="validatorVariables">
                    <var-name>datePattern</var-name>
                    <var-value>dateOfBirth.pattern</var-value>
                </var>

            </field>
        </form>
    </formset>
```

N.B. Validator provides the facility to indicate that a variable's value is in a resource bundle - but it is down to individual configured validators to actually take notice of these properties and retrieve the value from the resource bundle. Struts 1.3 has been upgraded to take advantage this feature and is an example of what custom validators need to do. See the Struts FieldChecks class.

1.3 Arg0-Arg3 Deprecations Removed

A number of deprecations have been removed in Validator 1.2.0 - however probably the biggest impact will be the **Arg0-Arg3** elements (which were replaced by **Arg**).

So for example, in the validator config...

```
        <arg0 key="first.key">
        <arg1 key="second.key">
```

should be changed to

```
        <arg key="first.key"  position="0">
        <arg key="second.key" position="1">
```

Position is however *optional* and if omitted, validator will automatically allocate a *position* value - see Bug 31194 for details.

Additionally, this issue only arises if you upgrade to the new version of the Validator DTD - previous versions of the DTD using arg0-arg3 will continue to work in Validator 1.2.0.

1.4 Changes to Validators

A few new validators have been added:

- A new **ISBN Validator** for validating book numbers has been added (see #31489).
- New *locale aware* **byte**, **short**, **integer**, **long**, **float** and **double** validations added (see #21282).
- New **Minimum/Maximum** value validators added (see #29015).

The following validator's issues have been fixed:

- **Email** Validator fixed for Bug 23990, Bug 29541, Bug 33409, Bug 31644
- **URL** Validator fixed for Bug 30686
- **Float** Validator fixed for Bug 32351 and Bug 36878
- **Double** Validator fixed for Bug 36878
- **Credit Card** Validator fixed for Bug 35926
- **Date** Validator no longer logs the ParseException for an invalid date.

The following **JavaScript** validator's issues have been fixed:

- The Strategy for locating a form using name/id has changed to correct various issues - see Bug 35127, Bug 35294 and Bug 31534
- **Required** validator fixed for Bug 15912 and Bug 33047
- **Date** validator enhanced to support *datePattern* - see Bug 22384

1.5 Dependency Changes

The following changes to validator's dependencies have occured:

- Commons BeanUtils - upgraded from version 1.6.1 to version 1.7.0
- Commons Collections - dependency removed (**see note below**).
- Commons Digester - upgraded from version 1.5 to version 1.6

- Commons Logging - upgraded from version 1.0.3 to version 1.0.4

N.B. Commons BeanUtils 1.7.0 removed its dependency on Commons Collections by including the few required Collections classes in its distribution. This is why Validator 1.2.0 no longer has a dependency on Collections - Collection's classes used by Validator are in BeanUtils 1.7.0.

---

2. Upgrading

This section describes upgrading from validator 1.1.4.

2.1 Deploying jars

Obvious, but don't forget to deploy the new Validator 1.2.0 jar and also upgrade to new versions of dependency jars for BeanUtils, Digester and Logging (see section 1.5 above).

2.2 New DTD version

There is a new version of the validator DTD: validator_1_2_0.dtd. However the previous version of the DTD (validator_1_1_3.dtd) will continue to work and there is no requirement to upgrade.

Upgrading your system to use validator_1_2_0.dtd has the following consequences:

- You will be able to use the new features that depend on the attributes that have been added to the DTD.
- If your validation.xml files uses the ar0-arg3 elements - these will need to be replaced with "arg" - see the section **1.3 Arg0-Arg3 Deprecations Removed** above.

NOT Upgrading your system to use validator_1_2_0.dtd has the following consequences:

- You won't be able to use (some of) the new features which depend on new elements in the DTD.

2.3 Struts Users

2.3.1 Struts 1.2.x

- Validator 1.2.0 can be deployed with Struts 1.2.x to take advantage of the new features(except one, see below).
- In Struts 1.2.7 Javascript validation doesn't work in XHTML mode (see Bug 35127) - resolving this bug requires upgrading to Struts 1.2.8 (hopefully soon to be released) and Validator 1.2.0
- The **Variables in Resource Bundles** feature described in section 1.2 above does not work in Struts 1.2.x - this requires Struts 1.3

2.3.2 Struts 1.3

Struts 1.3.5 is dependant on Validator 1.3.0 and fully supports all the fetures described above.