

# Contents

- [Upgrading Your Rules Definition Files](#)
  - [Changes to the Param Element](#)
  - [What Do I Have to Change?](#)
  - [Upgrading Your Files Using Eclipse](#)

## Upgrading Your Rules Definition Files

The format required for [rules definition files](#) changed in ValidateThis 0.98, in a way that makes older rules definition files incompatible. If you are upgrading from a version prior to 0.98 please read this page for more information about the change and a tip for easily upgrading your files using Eclipse.

### Changes to the Param Element

The only piece of metadata that changed its format is the *param* element. It is now a requirement to specify a *name* and *value* attribute for the *param* element, whereas prior to version 0.98 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 Do I 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.

## Upgrading Your Files Using Eclipse

Marc Esher was kind enough to contribute a regular expression that can be used with Eclipse's find and replace feature. In the dialog, using the following values:

**find:** <param (.\*)="(.)"

**replace:** <param name="\$1" value="\$2"

Make sure the "Regular Expressions" checkbox is checked.