08 ValidationRulesets

What	Required	Description	schema location
ValidationRulesets.xml	no	Definition of validation rule sets	https://fewsdocs.deltares.nl/schemas/version1.0/validationRuleSets.xsd

- Validation on extreme values
- Validation on rate of change
- Validation on series of same readings
- Validation on Temporary Shifts
- Validation on Oscillation
- flagSources
- Use of <timeSeries> instead of <timeSeriesSets>
- Examples of validation rules
- Changing validation

Validation rules are defined in DELFT-FEWS to allow quality checking of all time series data (scalar time series only). Several validation criteria may be defined per time series. All validation rules for all time series are defined in this configuration. For each time series to be checked, a set of validation rules is defined. Defining validation rules to apply to a time series set using a locationSet rather than identifying series individually can simplify the configuration greatly. Most validation rules may be defined either as a constant value, or as a value valid per calendar month.

When available on the file system, the name of the XML file for configuring the Validation Rule Sets is for example:

ValidationRuleSets 1.00 default.xml
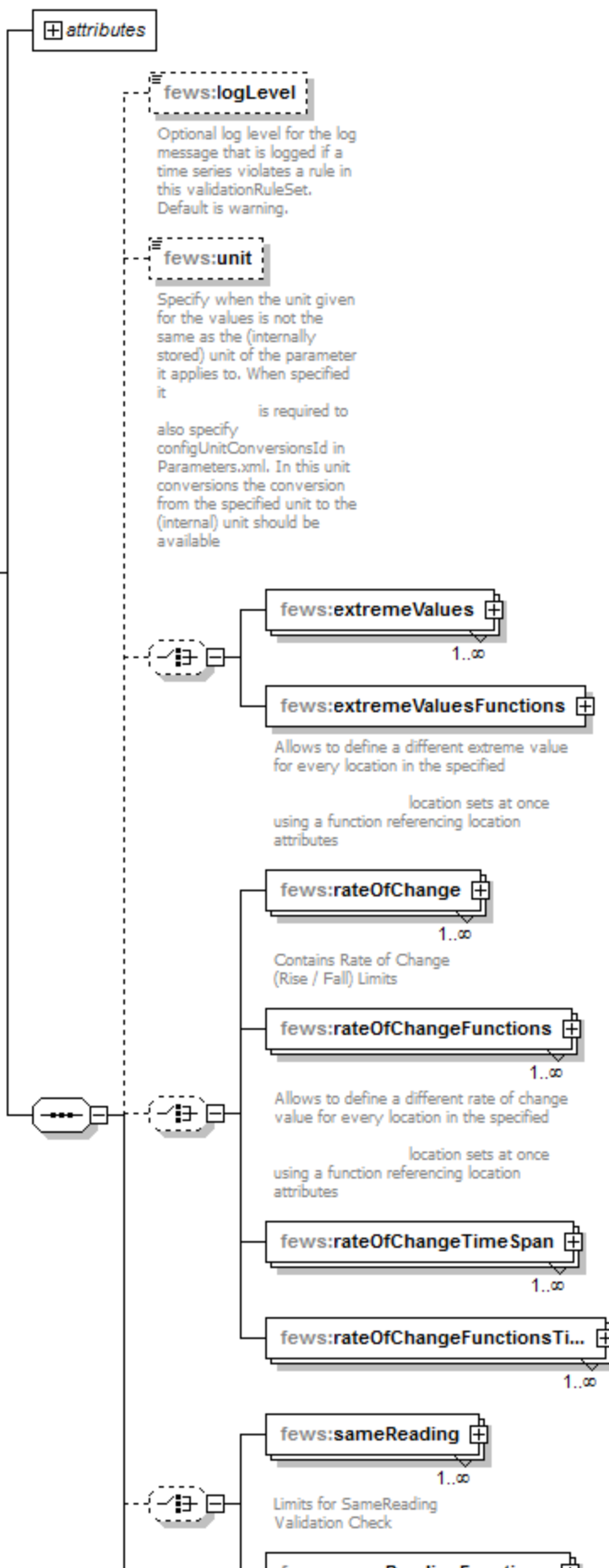
ValidationRuleSets   Fixed file name for the Validation rules configuration

1.00   Version number

default   Flag to indicate the version is the default configuration (otherwise omitted).

ValidationRuleSetComplexType

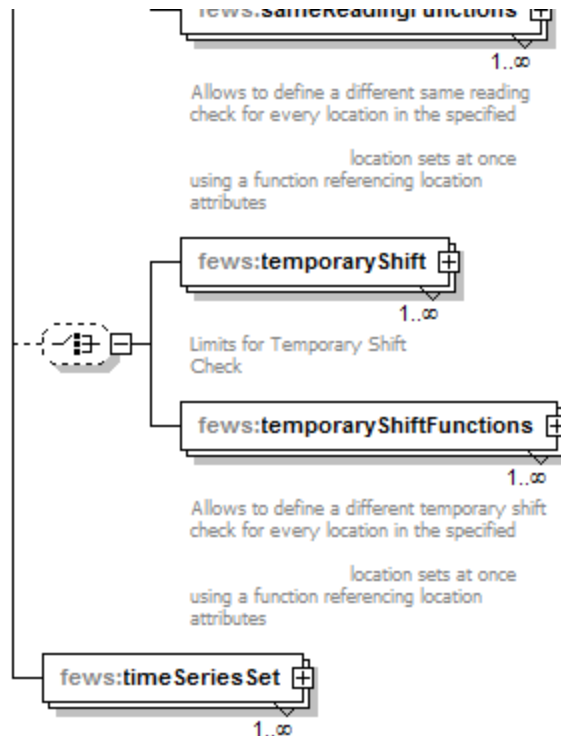Contains a Validation Rules Set for for a given time series

⊞ *attributes*

fews:logLevel

Optional log level for the log message that is logged if a time series violates a rule in this validationRuleSet. Default is warning.

fews:unit

Specify when the unit given for the values is not the same as the (internally stored) unit of the parameter it applies to. When specified it
 is required to also specify configUnitConversionsId in Parameters.xml. In this unit conversions the conversion from the specified unit to the (internal) unit should be available

fews:extremeValues
1..∞

fews:extremeValuesFunctions

Allows to define a different extreme value for every location in the specified
 location sets at once using a function referencing location attributes

fews:rateOfChange
1..∞

Contains Rate of Change (Rise / Fall) Limits

fews:rateOfChangeFunctions
1..∞

Allows to define a different rate of change value for every location in the specified
 location sets at once using a function referencing location attributes

fews:rateOfChangeTimeSpan
1..∞

fews:rateOfChangeFunctionsTi...
1..∞

fews:sameReading
1..∞

Limits for SameReading Validation Check

fews:sameReadingFunctions

fews:sameReadingFunctions

1..∞

Allows to define a different same reading check for every location in the specified

location sets at once using a function referencing location attributes

fews:temporaryShift

1..∞

Limits for Temporary Shift Check

fews:temporaryShiftFunctions

1..∞

Allows to define a different temporary shift check for every location in the specified

location sets at once using a function referencing location attributes

fews:timeSeriesSet

1..∞

Figure 33 Elements of the ValidationRuleSets configuration.

validationRuleSet

Root element of the definition of a validation rule set. Multiple entries may exist.

Attributes;

- validationRuleSetId: Optional reference ID for the validation rule set. Used only in messaging.
- timeZone: Shift (in hours) of the time zone to be used in considering time values in validation.

considerQualifiers

When matching timeSeriesSets during primary validation qualifiers are per default not considered. But by adding a <considerQualifiers> element to the validationRuleSet and setting it to true, the qualifiers also have to match for the validationRuleSet to match.

logLevel

Optional log level for the log message that is logged if a time series violates a rule in this validationRuleSet. Can be WARN, INFO or DEBUG. Default is WARN.

unit

Specify when the unit given for the values is not the same as the (internally stored) unit of the parameter it applies to. When specified it is required to also specify configUnitConversionsId in Parameters.xml. The conversion from the specified unit to the (internal) unit should be available in the unit conversions config file.

timeSeriesSet

Definition of the time series to apply validation rule to.

extremeValues

Validation rules defined to check for extreme values (hard and soft limits)

rateOfChange

Validation rules defined to check rate of change. Please note the units are per second i.e. 2m in 15mins is 0.00222.

sameReading

Validation rules defined to check for series of same readings.

temporaryShift

Validation rules defined to check for temporary shifts in time series.

These function have to do with Shape-DBF file configuration. See here

Location dependency

For extremeValues, rateOfChange, sameReading and temporaryShift a locationId can be given to make the rule location specific. Before 2014.02 this functionality was used to make multiple location-specific rules of the same type within a single validationRuleSet.

From 2014.02 on multiple rules of the same type within a single validationRuleSet can be given to apply to all locations within the timeSeriesSet instead of them to be location specific. Also from 2014.02 on multiple rateOfChangeFunctions, sameReadingFunctions and temporaryShiftFunctions can be used within the same validation rule set.

Validation on extreme values

This group of validation rules checks that the values in the time series do not exceed minimum and maximum limits. These limits may be defined as soft limits or as hard limits. Values exceeding soft limits will be marked as doubtful but retained. Values exceeding hard limits will be marked as unreliable. In case any of these rules are violated, a violation comment is added to the time step. If the time step already includes a comment, the violation comment is appended to the original comment. If no violation comment is provided, then the original comments would be added in the output only.
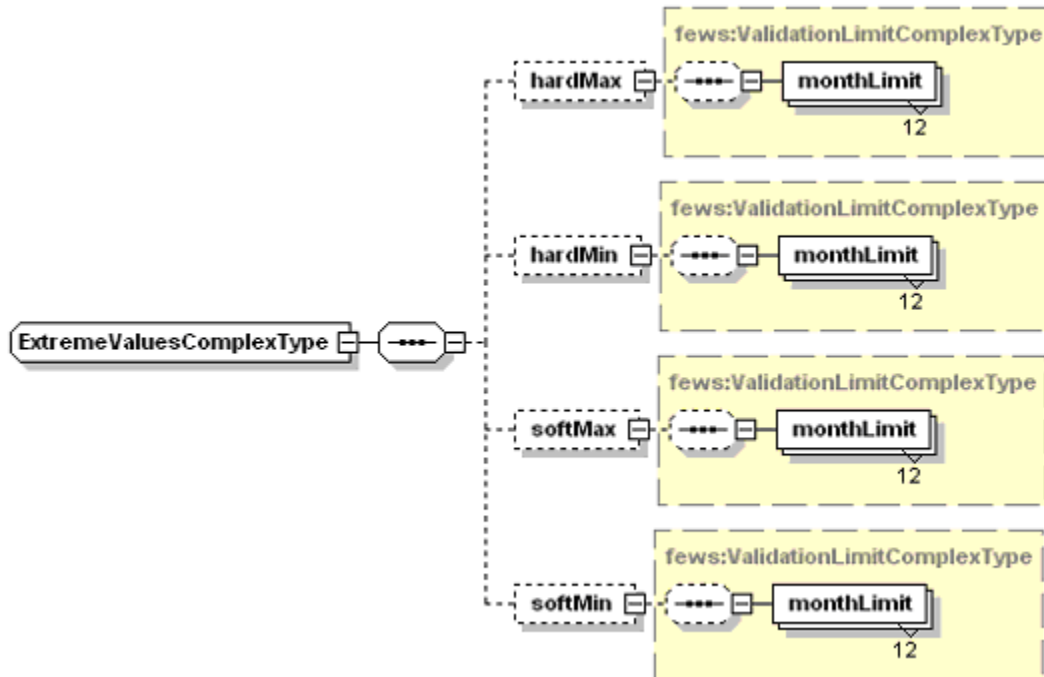

Figure 34 Elements of the Extreme values configuration of the ValidationRuleSets.

hardMax

Validation rule for checking hard maximum. Values exceeding this limit will be marked as unreliable.

Attributes;

- constantValue: Value of hardMax limit, used irrespective of time of value.

hardMin

Validation rule for checking hard minimum. Values exceeding this limit will be marked as unreliable.

Attributes;

- constantValue: Value of hardMin limit, used irrespective of time of value.

softMax

Validation rule for checking soft maximum. Values exceeding this limit will be marked as doubtful.

Attributes;

- constantValue: Value of softMax limit, used irrespective of time of value.

softMin

Validation rule for checking soft minimum. Values exceeding this limit will be marked as doubtful.

Attributes;

- constantValue: Value of softMin limit, used irrespective of time of value.

monthLimit

Element used when defining variable limits per calendar month. Twelve values must be defined. When defined the monthly limit will overrule the constant limit.

Validation on rate of change

This group of validation rules checks that the values in the time series do not exceed maximum rates of change. When the rate of change limit is exceeded, the values causing the limit to be exceeded will be marked as unreliable. Rate of change limits may be defined to be the same for the rate of rise as for the rate of fall. These may also be defined to be different. The rates need to be specified in the unit of the timeseries it applies per second. E.g. if you define a rate of change for a water level gauge with values in metres the rate should be given in metres per second. Please note that this validationRule can only be succesfully applied to a value if there is a value at the time step before it as well. Especially when applying this rule to a temporary series that is merged (i.e. including the validation flags) to a permanent series this can lead to unwanted behaviour. The first value of the temporary series can't be validated with this rule, as there exists no value before it, even though there is in fact a value present in the destination series of the merge.

If these rules are violated, and there exists an original comment for a given time step, no violation comment is added to the time step. However, if the time step does not include a comment, in the case of violating these rules, a violation comment is added to the time step.
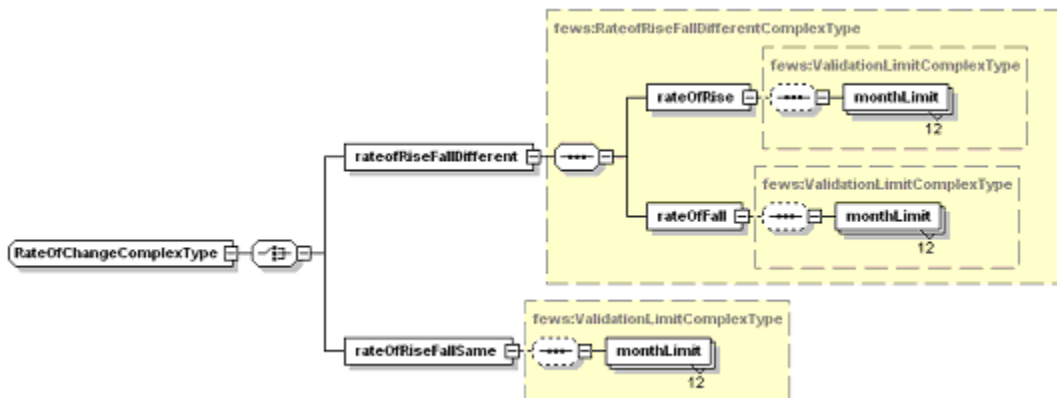

Figure 35 Elements of the rate of change configuration of the ValidationRuleSets.

rateofRiseFallDifferent

Root element used if the rate of rise limit is defined different to the rate of fall.

rateOfRise

Validation rule defined for the rate of rise.

Attributes;

- constantValue: Maximum rate of rise, used irrespective of date of the value. []

rateOfFall

Validation rule defined for the rate of fall.

Attributes;

- constantValue: Maximum rate of fall, used irrespective of date of the value. []

monthLimit

Element used when defining variable limits per calendar month. Twelve values must be defined. When defined the monthly limit will overrule the constant limit.

> ⓘ **RateOfChangeTimespan**
>
> For information on RateofChangeTimespan, see [this wiki page](#) (needs to be integrated in this page)

Validation on series of same readings

Time series of data can be validated on series of same readings. This may be unlikely for field observations, and may indicate an instrumental error. In some cases a small variability may still be observed, despite instrumental error. The same readings check allows for defining a bandwidth within the value is considered to be the same. Please note that this validationRule can only be succesfully applied to a value if there is a value at the time step before it as well. Especially when applying this rule to a temporary series that is merged (i.e. including the validation flags) to a permanent series this can lead to unwanted behaviour. The first value of the temporary series can't be validated with this rule, as there exists no value before it, even though there is in fact a value present in the destination series of the merge.

attributes

locationId

Specify a locationId to make the sameReading applicable to a single location only. When no locationId is specified, the sameReading will be applicable to all locations contained in the timeSeriesSets for this validationRuleSet.

validationFlag

Flag that will be set for all data values for which the configured limit is exceeded. Default is unreliable.

fews:label

Since 2013.01. FEWS-8621. Label that will be set for all data values for which the configured deviation and the reading period is exceeded. No label will be set by default

fews:comment

Since 2013.01. FEWS-8621. Comment that will be set for all data values for which the configured deviation and the reading period is exceeded. No label will be set by default

SameReadingComplexType

Limits for SameReading Validation Check

fews:sameReadingDeviaton

Contains Same Reading Deviation

fews:ValidationLimitComplexType

attributes

constantLimit

fews:monthLimit

12

fews:sameReadingPeriod

'constantLimit' and 'value' specify a period in seconds

fews:ValidationLimitComplexType

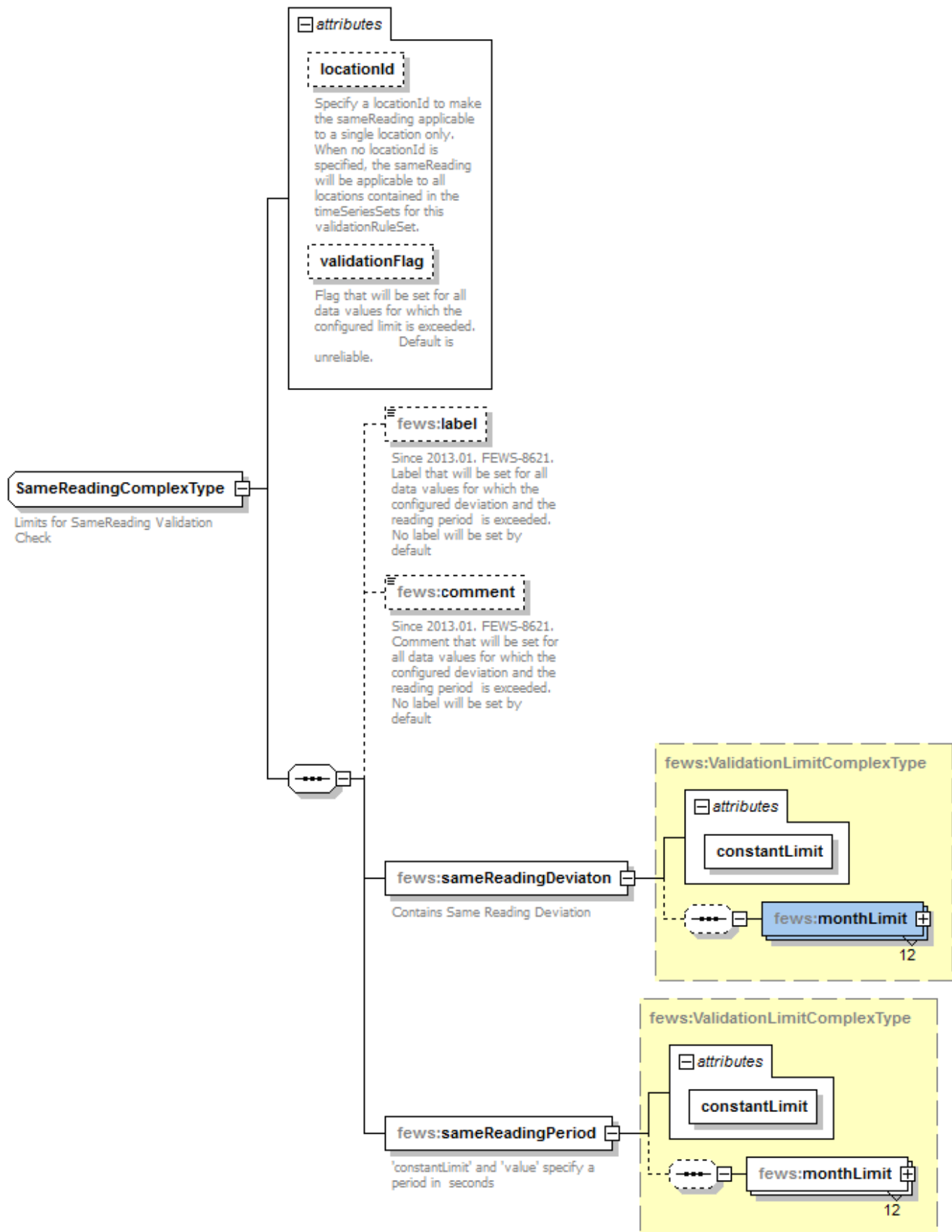**attributes**

constantLimit

fews:monthLimit

12

Figure 36 Elements of the same reading configuration of the ValidationRuleSets.

SameReadingViolations are marked as such if values stay within the sameReadingDeviation for at least the sameReadingPeriod. The way the check is applied is by having each value in the timeSeries be considered as potential start of a period in which the reading will be the same. In the case of a violation, a violation comment is added to the time step. If the time step already includes a comment, the violation comment is appended to the original comment.

For each non missing value (start value) a check on the timeSeries is applied as follows. After the start value, each non missing value is compared to the start value to see if the absolute difference is no more than the sameReadingDeviation. If it's equal to or less than the sameReadingDeviation, the value is considered to be within the bandwidth and the check is continued. If not (or if it was the last non missing value of the timeSeries), the check is stopped. If the time between the start value and the last value inside the bandwidth is equal to or greater than the sameReadingPeriod, all non missing values from the start value up to (and including) the last value are marked sameReading (SR).

After this, the check is applied to the first non missing value after, what was considered in the last check as, the starting value (even if it is already marked as sameReading).

sameReadingDeviation

Root element for definition of bandwidth the value may vary within if it is considered to the same reading. The bandwidth is twice the deviation.

Attributes;

- constantValue: Value for deviation (inclusive), used irrespective of date of the value.

sameReadingPeriod

Root element for definition of time span limit the value may remain the same to be considered realistic. If the reading remains the same for a longer period of time, ensuing values will be considered unreliable.

Attributes;

- constantValue: Value for time span in seconds (exclusive), used irrespective of date of the value.

monthLimit

Element used when defining variable limits per calendar month. Twelve values must be defined. When defined the monthly limit will overrule the constant limit.

excludeMissingsFromSameReadingPeriod

Since 2020.01. If true a same reading period ends when a missing value is found. Default is false.

Validation on Temporary Shifts

Time series of data can be validated on temporary shifts. These occur when instruments are reset, and can be identified by the values rapidly falling to a constant value, remaining at that value for a short period of time and then returning to the original value range. A complex set of validation criteria include the rate of change as well as a maximum time the value remains the same. Please note that this validationRule can only be succesfully applied to a value if there is a value at the time step before it as well (or multiple). Especially when applying this rule to a temporary series that is merged (i.e. including the validation flags) to a permanent series this can lead to unwanted behaviour. The first value of the temporary series can't be validated with this rule, as there exists no value before it, even though there is in fact a value present in the destination series of the merge.

In case of violating this rule, a violation comment is added to the time step. In that case, if a comment originally exists in the time step, it will be overridden by the violation comment.

TemporaryShiftComplexType

Limits for Temporary Shift Check

attributes

locationId

Specify a locationId to make the sameReading applicable to a single location only. When

no locationId is specified, the temporaryShiftPeriod will be applicable to all locations contained in the

timeSeriesSets for this validationRuleSet.

validationFlag

Flag that will be set for all data values for which the configured limit is exceeded. Default is unreliable.

fews:label

Since 2013.01. FEWS-8621. Label that will be set for all data values for which the configured deviation and the reading period is exceeded. No label will be set by default

fews:comment

Since 2013.01. FEWS-8621. Comment that will be set for all data values for which the configured deviation and the reading period is exceeded. No label will be set by default

fews:rateOfTemporaryShift

Contains rate of temporary shift (similar to rate of change)

fews:ValidationLimitComplexType

attributes

constantLimit

fews:monthLimit

12

fews:temporaryShiftPeriod

'constantLimit' and 'value' specify a period in seconds

fews:ValidationLimitComplexType

attributes

constantLimit
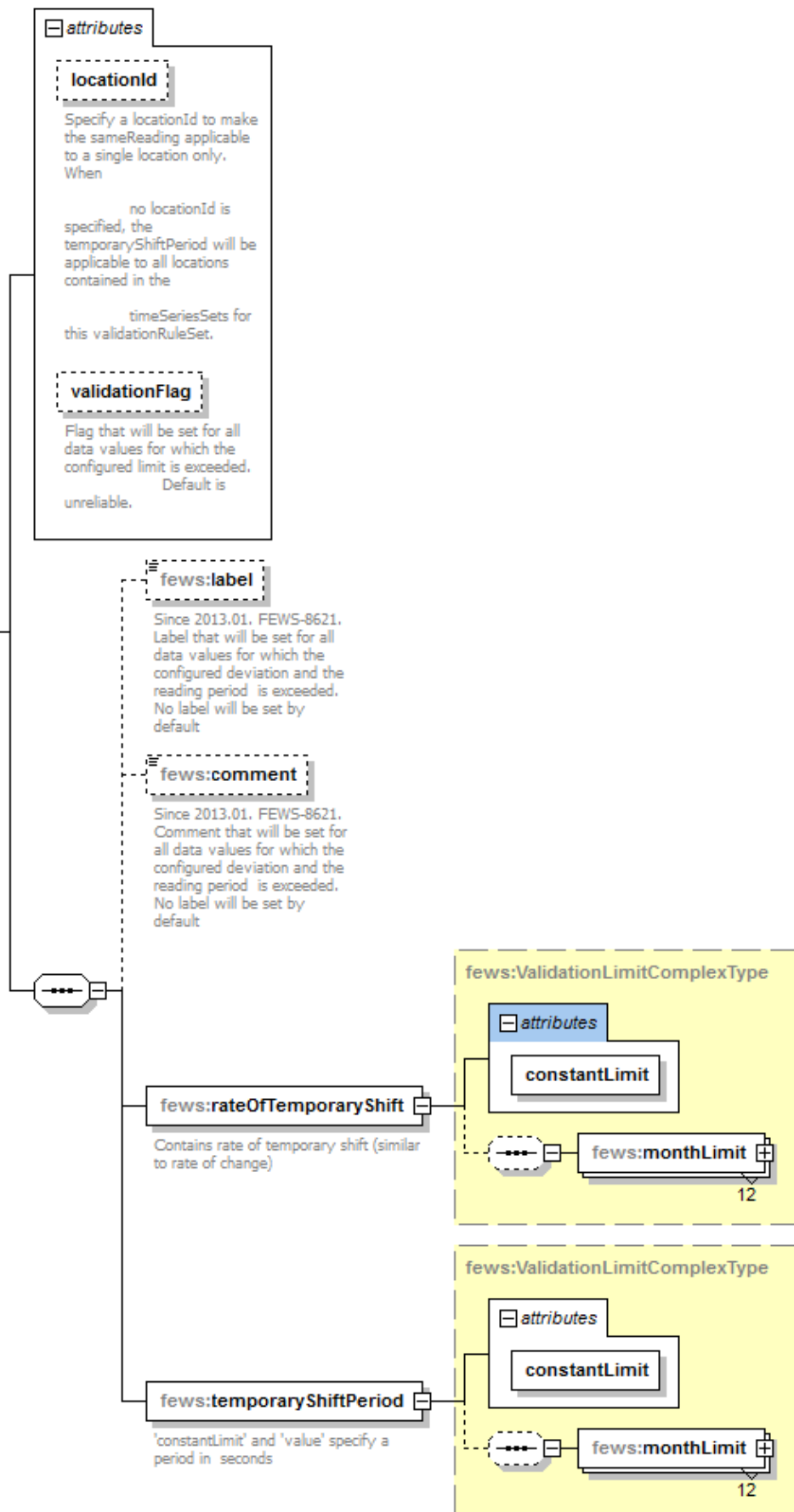
fews:monthLimit

12

Figure 37 Elements of the temporary shift configuration of the ValidationRuleSets.

rateOfTemporaryShift

Rate of change that must be exceeded both on change to shifted value and change back to original value range for validation rule to apply.

Attributes;

- constantValue: Value for rate of change, used irrespective of date of the value.

temporaryShiftPeriod

Maximum time span constant shifted value is in time series for validation rule to apply.

Attributes;

- constantValue: Value for time span in seconds, used irrespective of date of the value.

monthLimit

Element used when defining variable limits per calendar month. Twelve values must be defined. When defined the monthly limit will overrule the constant limit.

Validation on Oscillation

Time series can be validated on whether oscillation occurs. Oscillation is recognized when the value rapidly falls and rises in an alternating pattern. This validation can also be used to only mark values where oscillation occurs with an oscillation flag source, but keep the original flag of the value. This can be useful since oscillation might for example occur due to an error in the automatic steering of a pump causing it to rapidly turn on/off. In this case, the oscillation can be detected, without marking the actual values as "doubtful" or "unreliable" since the measured value is correct.

This can be configured through the <oscillation> element if you wish to specify constantValues or monthLimits for the various limits that need to be configured. The <oscillationFunctions> element can be used to specify the various limits using attributes.

minDifference

The minimum difference between a low and high value required for it to be detected as a potential oscillation.

maxPeriod

The maximum amount of time (in seconds) between two subsequent "low values" or two subsequent "high values" of a single oscillation
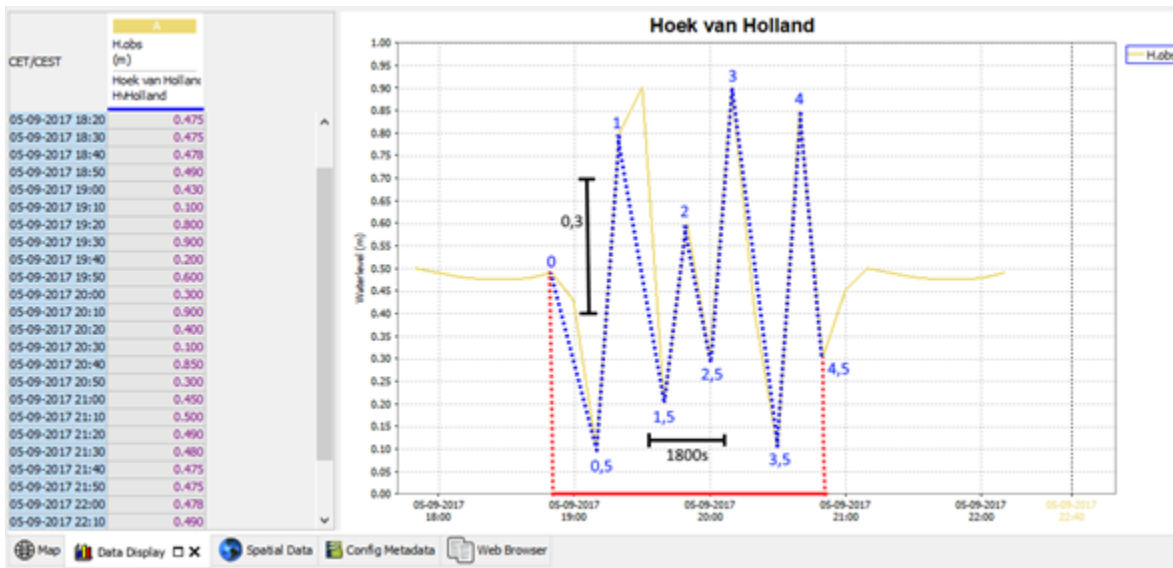
minOscillations

The minimum amount of oscillations required for the values to be flagged as oscillating. For example if the following pattern is found: high, low, high, low, this would be considered to be 1,5 oscillations.

validationFlag

Optional attribute to configure the validation flag that should be set for values in the oscillation period. When this validationFlag is not configured, only the oscillation flag source will be added to these values, and the flag will remain unchanged. Use this attribute when you want to set the flag of the values to doubtful or unreliable.

Example:

In the picture below 4.5 oscillations with a high vs low value difference of at least 0.3 and an oscillation period (2 subsequent high's or lows) of maximum 1800 seconds are illustrated.

Oscillation validation rule set example

```
<validationRuleSet validationRuleSetId="OscillationSimpleTest" timeZone="GMT">
    <oscillation validationFlag="doubtful">
        <minDifference constantLimit="0.3"/>
        <maxPeriod constantLimit="1800"/> <!-- 30 minutes -->
        <minOscillations constantLimit="2.5"/>
    </oscillation>
    <timeSeriesSet>
        ...
    </timeSeriesSet>
</validationRuleSet>
```

Oscillation validation rule set example using attributes

```
<validationRuleSet validationRuleSetId="OscillationAttributes" timeZone="GMT">
    <oscillationFunctions validationFlag="doubtful">
        <minDifference constantLimit="@MIN_DIFF@"/>
        <maxPeriod constantLimit="@MAX_PERIOD@"/>
        <minOscillations constantLimit="@MIN_OSC@"/>
    </oscillationFunctions>
    <timeSeriesSet>
        ...
    </timeSeriesSet>
</validationRuleSet>
```

flagSources

Since version 2012.01 FEWS stores not only the quality flags, but also the source of the flag, the so-called flagSource. So the user is able to see why a certain value is validated as unreliable, eg. due to exceeding of the hard max.
The list of flagSources is:

- IMP: flag is imported
- SN: soft min.
- HN: hard min.
- SX: soft max.
- HX: hard max.
- ROR: rate of rise
- ROF: rate of fall
- SR: same reading

- TS: temporary shift
- OSC: oscillation
- SC: secondary validation, series comparison
- FC: secondary validation, flag comparison
- KT: secondary validation, Mann-Kendall test
- MAN: manual edit

Example for "Rate of Rise" and "Temporary Shift" validation rules

Use of <timeSeries> instead of <timeSeriesSets>

Since 2023.02 it is possible to use the <timeSeries> element instead of the <timeSeriesSets> elements.

The <timeSeries> element works like a more general filter on all time series instead of the more explicitly defined <timeSeriesSet> element.

When there is no element defined for a time series key, there will be no filtering which means everything matches.

The most useful part of the <timeSeries> is the fact that multiple parameters can be defined instead of just the single one in a <timeSeriesSet>.

This can be done by defining multiple single parameters or even 1 or more parameter groups:

Example <timeSeries> in ThresholdValueSet

```
        <validationRuleSet validationRuleSetId="SetWithTimeSeriesFilter" timeZone="GMT">
                <extremeValues>
                        <hardMax constantLimit="20"/>
                        <hardMin constantLimit="0"/>
                        <softMax constantLimit="10"/>
                        <softMin constantLimit="2"/>
                </extremeValues>
                <timeSeries>
                        <moduleInstanceId>moduleA</moduleInstanceId>
                        <valueType>scalar</valueType>
                        <parameterGroupId>ParGroupA</parameterGroupId>
                        <parameterId>Par1</parameterId>
                        <parameterId>Par2</parameterId>
                        <locationSetId>locationSetA</locationSetId>
                        <timeSeriesType>external historical</timeSeriesType>
                        <timeStep unit="hour"/>
                </timeSeries>
        </validationRuleSet>
```

Examples of validation rules

Example for "Rate of Rise" and "Temporary Shift" validation rules

Changing validation

When validation rules have changed after data has been already validated, it is possible to revalidate all time series connected to all validation rules via the revalidation module which is available from version 2023.02.

When data in the past has been changed with validation rules like Same Reading, Temporary Shift or Oscillation connected to it, it is also advised to run this module.

This is because these validation rules work with a validation period which may not be enclosed entirely in the rewritten period. This could cause changes in validation.

If for example 100 identical subsequent values have been flagged as Same Reading, but then afterwards only the first 10 will be rewritten by a workflow with a relative view period in the past, those 10 values may not be flagged as Same Reading anymore because the values afterwards are not taken into account. The revalidation module, however, always revalidates the whole time series, and can be run for specific locations via the manual forecast dialog.