# FileDate constraints

## Summary

FileDate constraints check the last-modified property of the target resource. Checks are comparisons with literal strings or matching of string patterns or regular expressions.

## Element

<fileDate>

## Examples

<fileDate le="2020-07-08" leMsg="Files must be older than 2020-07-08"  
 gt="2020-04-01" gtMsg="Files must be from 2020-04-01 or later"/>

<fileDate like="2020\*" neMsg="File not from 2020"/>

<fileDate like="2020-05\*" neMsg="File not from May"/>

<fileDate like="2020-05-29" neMsg="File not from May 29 2020"/>

<fileDate like="\*T23\*" likeMsg="File not updated betwee 23:00 and 00:00"/>

<fileDate ne="2020-04-11T00:11:17.142Z" neMsg="This file version must not be used"/>

## Example schemas

See:

$greenfox/example-schemas/constraint-components/FileDate

## Constraint nodes

The constraints are specified by attributes on <fileDate>, summarized in the following table.

**Table 1**: Constraint nodes supported by FileDate constraints. All nodes are attribues on the <fileDate> element.

|  |  |  |  |
| --- | --- | --- | --- |
| **Check node** | **Semantics** | **Example** | **Notes** |
| @eq, @ne,  @lt, @le,  @gt, @ge | Last-modified date is eq|ne|lt|le|gt|ge the attribute value | eq="2020-08-11T09:39:47.552Z"  lt="2020-08-11T10:00"  gt="2020-08-11"  lt="2020-09" | Compares as strings, thus initial substrings like “2020-09” can be used |
| @like | Last-modified date matches the text pattern | like="2020-08-11T10\*" | Wildcards are \* and ? |
| @notLike | Last-modified date does not match the text pattern | notLike="2020-08-11\*" | Wildcards are \* and ? |
| @matches | Last-modified date matches the regex | matches="T(23|00)" | The regex is not anchored – it may describe a substring |
| @notMatches | Last-modified date does not match the regex | notMatches="T(05|06)" | The regex is not anchored – it may describe a substring |

## Option nodes

There is a single option node, represented in the following table.

**Table 2**: The only options node supported by FileDate constraints. The node is an attribute on the <fileDate> element.

|  |  |  |  |
| --- | --- | --- | --- |
| **Check node** | **Semantics** | **Example** | **Notes** |
| @flags | The attribute value supplies flags used when evaluating regular expressions or string patterns. | @flags="i"  @flags="x" | Flag semantics as described in the XPath functions spec ([XP flags](https://www.w3.org/TR/xpath-functions-31/#flags)); “x” can be useful when ignorable whitespace makes the regex more readible |

## Message nodes

For each constraint node (see table 1), two message nodes are supported – an error message and a message commenting a pass, as summarized in the following table.

**Table 3**: Message nodes of FileDate constraints. The nodes are attributes on the <FileDate> element.

|  |  |  |
| --- | --- | --- |
| **Message node** | **Semantics** | **Example** |
| constraint-node-name + „Msg“ | Error message, displayed in a red result describing constraint violation | gtMsg="File stale" |
| constraint-node-name + „MsgOK“ | Success message, displayed in a green result describing conformance | gtMsgOK="File fresh" |