# FileName constraints

## Summary

FileName constraints check the name of the target resource. Checks are comparison with literal strings or matching of string patterns or regular expressions.

## Element

<fileName>

## Examples

<fileName like="airport\*.json" likeMsg="JSON files must match pattern 'airport\*.json"  
 notLike="\*test\*" notLikeMsg="JSON files must not match pattern '\*test\*'"/>

<fileName notMatches="\s"

likeMsg="Billions of dollars are wasted by using whitespace in file names"/>

## Example schemas

See:

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

## Constraint nodes

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

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Check node** | **Semantics** | **Example** | **Notes** |
| @eq, @ne | The name is eq or ne the attribute value | eq="airports.xml"  ne="UPDATE\_FLAG" | Wildcards not allowed – use @like or @notLike in order to use wildcards |
| @like | The name matches the text pattern | like="\*.xsd"  notLike="\*.deprecated.\*" | Wildcards are \* and ? |
| @notLike | The name does not match the text pattern | notLike="\*.deprecated.\*" | Wildcards are \* and ? |
| @matches | The name matches the regex | matches="\s\d+" | The regex is not anchored – it may describe a substring |
| @notMatches | The name does not match the regex | notMatches="\(.\*\)" | The regex is not anchored – it may describe a substring |

## Option nodes

There are two option nodes, represented in the following table.

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Check node** | **Semantics** | **Example** | **Notes** |
| @case | If true, the comparison is case-sensitive | case="true" | Comparison is case-sensitive. By default, it is case-insensitive, regardless if the comparisons uses literal strings (@eq, @ne) or string patterns or regular expressions. |
| @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 FileName constraints. The nodes are attributes on the <fileName> 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" |