## Information model

This section describes the information model underlying the operations of greenfox.

### Part 1: resource model

A **file system tree** is a tree whose nodes are resources – folders and files.

A resource has an identity, resource properties, derived resource properties and resource values.

The **resource identity** of a file system resource can be expressed by a combination of file system identity and a file path within the file system.

A **resource property** has a name and a value which can be represented by an XDM value.

A **derived resource property** is a property of a resource property value, or of a derived resource property value, which can be represented by an XDM value.

A **resource value** is a resource property value, a derived resource property value, or the value of another mapping of a resource to an XDM value.

#### Folder resources

**Table x.x. Resource properties of a folder resource, as currently evaluated by greenfox.** More properties may be added, e.g. representing access rights or a SHA-1 value.

|  |  |  |
| --- | --- | --- |
| **Property name** | **Value type** | **Description** |
| [name] | xsd:string | The folder name; optional – the file system root folder does not have a name |
| [parent] | Folder resource | The XDM representation identifying the parent is its file path |
| [children] | Folder and File resources | The XDM representation identifying the children are their file paths |
| [last-modified] | xsd:dateTime | May be out of sync when comparing values of resources from different machines |

A folder has the following **derived resource properties**:

**Table x.x. Derived resource properties of a folder resource, as currently evaluated by greenfox.**

|  |  |  |
| --- | --- | --- |
| **Property name** | **Value type** | **Description** |
| [filepath] | xsd:string | The names of all ancdestor folders and the folder itself, separated by a slash |
| [foxpath-value] | Mapping:  foxpath expression string =>  XDM value | A mapping of foxpath expressions to an XDM value, which is the value obtained by evaluating the expression in the context of the resource folder’s [filepath] value |

#### File resources

A file has the following **resource properties**:

**Table x.x. Resource properties of a file resource, as currently evaluated by greenfox.** More properties may be added, e.g. representing access rights or a SHA-1 value.

|  |  |  |
| --- | --- | --- |
| **Property name** | **Value type** | **Description or remark** |
| [name] | xsd:string | Optional – the file system root folder does not have a name |
| [parent] | Folder resource | The XDM representation identifying the parent is its file path |
| [text] | xsd:string | The text content of the file (empty if not a text file) |
| [encoding] | xsd:anyURI | The encoding of the text content of the file (empty if not a text file) |
| [octets] | xsd:base64Binary | The binary file content |
| [xmldoc] | document-node() | The result of parsing [text] into an XML document |
| [jsondoc-basex] | document-node() | The result of parsing [text] into a JSON document represented by a document-node in accordance with BaseX documentation |
| [jsondoc-w3c] | document-node() | The result of parsing [text] into a JSON document represented by a document-node in accordance with XPath function fn:json-to-xml |
| [htmldoc] | document-node() | The result of parsing [text] into an XML document represented by a document-node in accordance with the rules defined by TagSoup |
| [csvdocs] | Mapping:  csv-parse-parameters =>  document-node() | The mapping result is a CSV document represented by a document-node as controlled by the parsing parameters, in accordance with the BaseX documentation, |
| [last-modified] | xsd:dateTime | May be out of sync when comparing values of resources from different machines |
| [size] | xsd:integer | File size, in bytes |

A file has the following **derived resource properties**.

**Table x.x. Derived resource properties of a file resource, as currently evaluated by greenfox.** More properties may be added.

|  |  |  |
| --- | --- | --- |
| **Property name** | **Value type** | **Description** |
| [xmldoc-xpath] | Mapping:  xpath expression string =>  XDM value | A mapping of XPath expressions to an XDM value, which is the value obtained by evaluating the expression in the context of [xmldoc] |
| [jsondoc-basex-xpath] | Mapping:  xpath expression string =>  XDM value | A mapping of XPath expressions to an XDM value, which is the value obtained by evaluating the expression in the context of [jsondoc-basex] |
| [jsondoc-w3c-xpath] | Mapping:  xpath expression string =>  XDM value | A mapping of XPath expressions to an XDM value, which is the value obtained by evaluating the expression in the context of [jsondoc-w3c] |
| [htmldoc-xpath] | Mapping:  xpath expression string =>  XDM value | A mapping of XPath expressions to an XDM value, which is the value obtained by evaluating the expression in the context of [htmldoc] |
| [csvdoc-xpath] | Mapping:  (  csv-parse-parameters,  XPath expression string  ) =>  XDM value | A mapping of csv-parse-parameters and an XPath expression to an XDM value, which is the value obtained by evaluating the expression in the context of a document node from [csv-docs], obtained for the csv-parse-parameters |
| [foxpath-value] | Mapping:  foxpath expression string =>  XDM value | A mapping of foxpath expressions to an XDM value, which is the value obtained by evaluating the expression in the context of [filepath] |

Mapping [text] to CSV documents is controlled by csv-parse-params, see appendix x.x.

### Part 2: schema model

File system validation is a mapping of a file system tree and a greenfox schema to a set of greenfox validation results.

A **greenfox schema** is a set of shapes.

A **shape** is a resource shape or a value shape.

A **resource shape** is a set of constraints and an optional target declaration.

A resource shape is a **folder shape** or a **file shape**.

A **target declaration** specifies the selection of a target.

A **target** is a set of focus resources.

A **focus resource** is a resource to be validated against a resource shape.

A **value shape** is a mapping of a focus resource to a resource value and a set of constraints.

A **constraint** maps a resource value or a particular resource property to a boolean value.

A constraint identifies a constraint component and assigns values to the constraint component parameters.

A **constraint component** is a set of parameter declarations and a validator.

A **parameter declaration** specifies the name and type of a mandatory or optional parameter used by a validator.

A **validator** is a set of rules how a resource value and the arguments bound to the parameters are mapped to a validation result.

*An* ***observer*** *is a set of rules how a resource and a subset of arguments are mapped to an observation.*

### Part 3: validation model

*File system validation* is a mapping of a file system and a greenfox schema to a set of greenfox validation results.

Validation of a file system tree against a greenfox schema: Given a file system tree and a greenfox schema, the validation results are the union of results of the validation of the file system tree against all shapes in the greenfox schema.

Validation of a file system tree against a shape: Given a file system tree and a shape in the greenfox schema, the validation results are the union of the results of the validation of each resource which is in the target of the shape of the schema.

Validation of a resource against a shape: Given a resource in the file system tree and a shape in the greenfox schema, the validation results are the union of the results of the validation of the resource against all constraints declared by the shape, unless the shape has been deactivated, in which case the validation results are empty.

Validation of a focus resource against a constraint: Given a resource in the file system tree and a constraint of kind C in the greenfox schema, the validation results are defined by the validators and observers of the constraint component C. These validators and observers typically take as input a resource value (e.g. [xmldoc]) of the resource and the arguments supplied to the constraint component parameters.

## Appendix Ax: csv parsing parameters

The following parameters (collectively called csv parsing parameters) control the parsing of a CSV file into an XML document:

* separator : xs:string; default: “,”
* header: xs:boolean (false); default: false
* names-format: xs:string (one of “elementName”, “attributeValue”); default: elementName
* backslashes: xs:boolean; default: false
* quote: xs:boolean; default: trie

(better: a table)

A greenfox schema is a set of shapes.

A **shape** associates a target with a set of perceptions.

A **target** is a set of resources.

A **perception** associates a resource value with a constraint or an observation.

A **resource value** is the result of mapping a resource to an XDM value.

A **constraint** is a mapping of a resource value to a boolean value expressing conformance.

A constraint identifies a constraint component and determines the values of constraint component parameters.

A **constraint component** defines a set of parameters and defines how a given resource or resource set is mapped to a boolean value, as controled by the parameter values.

An **observation** is a mapping of a resource value to itself or another value