|  |
| --- |
| The MITRE Corporation |
| The OVAL® Language Independent Component Model Specification |
| Version 5.10.1 |
|  |
| **Danny Haynes, Stelios Melachrinoudis** |
| **2/24/2012** |

|  |
| --- |
| The Open Vulnerability and Assessment Language (OVAL®) is an international, information security, community standard, being developed in collaboration with any and all interested parties, to promote open and publicly available security content, and to standardize the transfer of this information across the entire spectrum of security tools and services. By standardizing the three main steps of the assessment process: representing configuration information of systems for testing; analyzing the system for the presence of the specified machine state; and reporting the results of the assessment, the OVAL Language provides a common and structured format that facilitates collaboration and information sharing among the information security community as well as interoperability among tools. This document defines the independent OS data model for the OVAL Language. |

# Acknowledgements

# Trademark Information

OVAL and the OVAL logo are registered trademarks of The MITRE Corporation. All other trademarks are the property of their respective owners.

# Terms of Use

MITRE MAKES OVAL AVAILABLE ON AN "AS IS" BASIS AND THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE MITRE CORPORATION, ITS BOARD OF TRUSTEES, OFFICERS, AGENTS, AND EMPLOYEES, DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION THEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.[[1]](#footnote-1)

# Feedback

The MITRE Corporation welcomes any feedback regarding the OVAL Language UNIX Component Model Specification. Please send any comments, questions, or suggestions to the public OVAL Developer's Forum at [oval-developer-list@lists.mitre.org](mailto:oval-developer-list@lists.mitre.org) or directly to the OVAL Moderator at [oval@mitre.org](mailto:oval@mitre.org)[[2]](#footnote-2).

Contents

[Acknowledgements 1](#_Toc322096499)

[Trademark Information 1](#_Toc322096500)

[Terms of Use 1](#_Toc322096501)

[Feedback 1](#_Toc322096502)

[1. Introduction 3](#_Toc322096503)

[1.1 Document Conventions 3](#_Toc322096504)

[1.2 Document Structure 5](#_Toc322096505)

[2. OVAL Language Independent Component Model 5](#_Toc322096506)

[2.1 Data Model Conventions 5](#_Toc322096507)

[2.2 ind-def:environmentvariable\_test 5](#_Toc322096508)

[2.2.1 Known Supported Platforms 6](#_Toc322096509)

[2.3 ind-def:environmentvariable\_object 6](#_Toc322096510)

[2.4 ind-def:environmentvariable\_state 7](#_Toc322096511)

[2.5 ind-sc:environmentvariable\_item 7](#_Toc322096512)

[2.6. ind-def:family\_test 8](#_Toc322096513)

[2.6.1. Known Supported Platforms 9](#_Toc322096514)

[2.7. ind-def:family\_object 9](#_Toc322096515)

[2.8. ind-def:family\_state 9](#_Toc322096516)

[2.9. ind-sc:family\_item 9](#_Toc322096517)

[2.10. ind-def:EntityStateFamilyType 10](#_Toc322096518)

[2.11. ind-sc:EntityItemFamilyType 10](#_Toc322096519)

[2.12. ind-def:unknown\_test 11](#_Toc322096520)

[2.13 ind-def:textfilecontent\_test 11](#_Toc322096521)

[2.13.1 Known Supported Platforms 12](#_Toc322096522)

[2.14 ind-def:textfilecontent\_object 12](#_Toc322096523)

[2.15 ind-def:FileBehaviors 14](#_Toc322096524)

[2.16 ind-def:textfilecontent\_state 16](#_Toc322096525)

[2.17 ind-sc:textfilecontent\_item 18](#_Toc322096526)

[2.18. ind-def:EntityStateWindowsViewType 19](#_Toc322096527)

[2.19. ind-sc:EntityItemWindowsViewType 19](#_Toc322096528)

[2.20 ind-def:textfilecontent54\_test 21](#_Toc322096529)

[2.20.1 Known Supported Platforms 21](#_Toc322096530)

[2.21 ind-def:textfilecontent54\_object 21](#_Toc322096531)

[2.22 ind-def:Textfilecontent54Behaviors 24](#_Toc322096532)

[2.23 ind-def:textfilecontent54\_state 25](#_Toc322096533)

[2.24 ind-sc:textfilecontent54\_item 27](#_Toc322096534)

[2.25. ind-def:EntityStateWindowsViewType 30](#_Toc322096535)

[2.26. ind-sc:EntityItemWindowsViewType 30](#_Toc322096536)

[2.27 ind-def:variable\_test 31](#_Toc322096537)

[2.27.1 Known Supported Platforms 32](#_Toc322096538)

[2.28 ind-def:variable\_object 32](#_Toc322096539)

[2.29 ind-def:variable\_state 33](#_Toc322096540)

[2.30 ind-sc:variable\_item 33](#_Toc322096541)

[2.31. ind-def:EntityObjectVariableRefType 34](#_Toc322096542)

[2.32. win-def:EntityStateVariableRefType 34](#_Toc322096543)

[2.33. win-sc:EntityItemVariableRefType 35](#_Toc322096544)

[2.34 ind-def:xmlfilecontent\_test 36](#_Toc322096545)

[2.34.1 Known Supported Platforms 37](#_Toc322096546)

[2.35 ind-def:xmlfilecontent\_object 37](#_Toc322096547)

[2.36 ind-def:FileBehaviors 39](#_Toc322096548)

[2.37 ind-def:xmlfilecontent\_state 41](#_Toc322096549)

[2.38 ind-sc:xmlfilecontent\_item 42](#_Toc322096550)

[Appendix A – Normative References 45](#_Toc322096551)

[Appendix B - Change Log 45](#_Toc322096552)

[Appendix C – Terms and Acronyms 45](#_Toc322096553)

# Introduction

## 1.1 Document Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in *RFC 2119* [1].

The following font and font style conventions are used throughout the remainder of this document:

* The Courier New font is used for writing constructs in the OVAL Language Data Model and to indicate commands on the UNIX command line.

Examples: generator (OVAL Construct), ls –al (UNIX command)

* The *'italic, with single quotes'* font is used for noting values for OVAL Language properties.

Example: *'does not exist'*

* The bold font and the keyword **Default Value:** are used to indicate a property's default value.

Example: **Default Value: -1**

* The bold font and the keyword **xsi:nil="true":** are used to indicate the meaning of an entity when the xsi:nil property is set to true.

Example: **xsi:nil="true"** indicates that the file\_object MUST collect the set of directories specified by the path entity. In addition, a value, for the filename entity, MUST NOT be specified.

This document uses the concept of namespaces[[3]](#footnote-3) to logically group OVAL constructs throughout both the Data Model section of the document, as well as other parts of the specification. The format of these namespaces is prefix:element, where the prefix is the namespace component, and the element is the name of the qualified construct. The following table lists the namespaces used in this document:

|  |  |  |  |
| --- | --- | --- | --- |
| Data Model | Namespace | Description | Example |
| OVAL Definitions | oval-def | The OVAL Definitions data model that defines the core framework constructs for creating OVAL Definitions. This is defined in the OVAL Language Specification [2]. | oval-def:TestType |
| OVAL System Characteristics | oval-sc | The OVAL System Characteristics data model, which defines the constructs used to capture the data collected on a target system. This is defined in the OVAL Language Specification. | oval-sc:ItemType |
| Independent Definitions | ind-def | The Independent Definitions data model defines the platform-specific constructs used in OVAL Definitions to make assertions about the state of various systems. | ind-def:family\_test |
| Independent System Characteristics | ind-sc | The Independent System Characteristics data model defines the platform-specific constructs used in OVAL System Characteristics to represent the system state information collected from various systems. | ind-sc:family\_item |

Lastly, each OVAL Test will contain a section titled "Known Supported Platforms" that specifies which platforms the OVAL Test is known to work on. This section is provided for convenience only and should not be considered a comprehensive list. In addition, there may be further known support restrictions specified for behaviors or entities that supersede the "Known Supported Platforms" section for the OVAL Test.

## 1.2 Document Structure

This document serves as the specification for the independent extension of the OVAL Language Specification and defines the platform-specific data model. This document is organized into the following sections:

* Section 1 – Introduction
* Section 2 – OVAL Language Independent Component Model
* Appendix A – References
* Appendix B – Change Log
* Appendix C – Terms and Acronyms

# OVAL Language Independent Component Model

The OVAL Language Independent Component Data Model is the extension of the OVAL Language Data Model for various operating systems.

## Data Model Conventions

This document follows the data model conventions described in Section 4.1 of the OVAL Language Specification.

## ind-def:environmentvariable\_test

The environmentvariable\_test is used to make assertions about environment variable(s)[[4]](#footnote-4) found on the system. The environmentvariable\_test MUST reference one environmentvariable\_object and zero or more environmentvariable\_states.



### Known Supported Platforms

## ind-def:environmentvariable\_object

The environmentvariable\_object construct defines the specific environment variable(s)[[5]](#footnote-5) who characteristics should be collected and represented as environmentvariable\_items.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Type | Multiplicity | Nillable | Description |
| set | oval-def:set | 0..1 | false | Enables the expression of complex environmentvariable\_objects that are the result of logically combining and filtering the environmentvariable\_items that are identified by one or more environmentvariable\_objects.  Please see the OVAL Language Specification for additional information. |
| name | oval-def:  EntityObjectStringType | 0..1 | true | The name of an environmental variable to evaluate. |
| filter | oval-def:filter | 0..\* | false | Allows for the explicit inclusion or exclusion of environmentvariable\_items from the set of environmentvariable\_items collected by an environmentvariable\_object.  Please see the OVAL Language Specification [2] for additional information. |

## ind-def:environmentvariable\_state

The environmentvariable\_state construct is used by an environmentvariable\_test to specify the name and value of the environmental variable to be checked[[6]](#footnote-6).



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Type | Multiplicity | Nillable | Description |
| name | oval-def:EntityStateStringType | 0..1 | false | The name of an environment variable. |
| value | oval-def:  EntityStateAnySimpleType | 0..1 | false | The actual value of the specified environment variable. |

## ind-sc:environmentvariable\_item

The environmentvariable\_item construct defines the information associated with environment variables[[7]](#footnote-7).



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Type | Multiplicity | Nillable | Description |
| name | oval-sc:EntityItemStringType | 0..1 | false | The name of an environment variable. |
| value | oval-sc:  EntityItemAnySimpleType | 0..1 | false | The actual value of the specified environment variable. |

## ind-def:family\_test

The family\_test is used to make assertions about information associated with the OS family the machine is running on. The family\_test MUST reference one family\_object and zero or more family\_states.  


### Known Supported Platforms

## ind-def:family\_object

The family\_object construct defines the information about the machine's OS family that should be collected and represented as family\_items. Since there is only one object relating to the machine's OS family, there are no child entities defined for this object, so it is considered empty.



## ind-def:family\_state

The family\_state construct is used by a family\_test to specify OS family information on a machine.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Type | Multiplicity | Nillable | Description |
| family | ind-def:  EntityStateFamilyType | 0..1 | false | This property specifies the high-level system OS type to test against. Please refer to the definition of the EntityFamilyType for more information about the possible values. |

## ind-sc:family\_item

The family\_item construct specifies information about the OS family information on a machine.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Type | Multiplicity | Nillable | Description |
| family | ind-sc:  EntityItemFamilyType | 0..1 | false | This property specifies the high-level system OS type to test against. Please refer to the definition of the EntityFamilyType for more information about the possible values. |

## ind-def:EntityStateFamilyType

The EntityStateFamilyType defines the values that describe high-level families of OSes.

|  |  |
| --- | --- |
| Enumeration Value | Description |
| catos | Describes the Cisco CatOS operating system. |
| ios | Describes the Cisco IOS operating system. This is NOT the same as the iOS operating system that Apple uses for IPhones and IPads. |
| macos | Describes the Mac operating system. |
| pixos | Describes the Cisco PIX operating system. |
| undefined | Indicates that the desired family is not available. |
| unix | Describes the UNIX operating system. |
| vmware\_infrastructure | Describes the VMWare Infrastructure. |
| windows | Describes the Microsoft Windows operating system. |
| *<empty string>* | The empty string value is permitted here to allow for empty elements associated with variable references. |

## ind-sc:EntityItemFamilyType

The EntityItemFamilyType defines the values that describe high-level families of OSes.

|  |  |
| --- | --- |
| Enumeration Value | Description |
| catos | Describes the Cisco CatOS operating system. |
| ios | Describes the Cisco IOS operating system. This is NOT the same as the iOS operating system that Apple uses for IPhones and IPads. |
| macos | Describes the Mac operating system. |
| pixos | Describes the Cisco PIX operating system. |
| undefined | Indicates that the desired family is not available. |
| unix | Describes the UNIX operating system. |
| vmware\_infrastructure | Describes the VMWare Infrastructure. |
| windows | Describes the Microsoft Windows operating system. |
| *<empty string>* | The empty string value is permitted here to allow for empty elements associated with variable references. |

## ind-def:unknown\_test

The unknown\_test acts as a placeholder for tests whose implementation is unknown. Any information that is known about the test should be held in the notes child element that is available through the extension of the abstract test element. Also, the required check attribute that is part of the extended TestType should be ignored during evaluation and hence can be set to any valid value.



## ind-def:textfilecontent\_test

The textfilecontent\_test is used to make assertions on systems about the contents of text files by looking at individual lines. The textfilecontent\_test MUST reference one textfilecontent\_object and zero or more textfilecontent\_states.



### Known Supported Platforms

## ind-def:textfilecontent\_object

The textfilecontent\_object construct defines the set of text files whose associated information should be collected and represented as textfilecontent\_items. The textfilecontent\_object will only collect regular files[[8]](#footnote-8) on UNIX systems and FILE\_TYPE\_DISK[[9]](#footnote-9) files on Windows systems. It is STRONGLY RECOMMENDED that **binary files** are NOT used, as they can contain characters that are detrimental for parsing and XML compatibility.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Type | Multiplicity | Nillable | Description |
| set | oval-def:set | 0..1 | false | Enables the expression of complex textfilecontent\_objects that are the result of logically combining and filtering the textfilecontent\_items that are identified by one or more textfilecontent\_objects.  Please see the OVAL Language Specification for additional information. |
| behaviors | ind-def:FileBehaviors | 0..1 | false | Specifies the behaviors that direct how the textfilecontent\_object collects textfilecontent\_items from the system. |
| path | oval-def:  EntityObjectStringType | 0..1 | false | The directory component of the absolute path to a file on the machine. |
| filename | oval-def:  EntityObjectStringType | 0..1 | false | The name of the file. |
| line | oval-def:  EntityObjectStringType | 0..1 | false | The line element represents a line in the file and is represented using a regular expression. A single subexpression can be called out using parentheses. Note that when using regular expressions, OVAL supports a common subset of the regular expression character classes, operations, expressions and other lexical tokens defined within Perl 5's regular expression specification[[10]](#footnote-10). |
| filter | oval-def:filter | 0..\* | false | Allows for the explicit inclusion or exclusion of textfilecontent\_items from the set of textfilecontent\_items collected by a textfilecontent\_object.  Please see the OVAL Language Specification [2] for additional information. |

## ind-def:FileBehaviors

The FileBehaviors construct defines the behaviors that direct how files are collected from the system. Note that using these behaviors may result in some unique results. For example, a double negative type condition might be created where an object entity says include everything except a specific item, but a behavior is used that might then add that item back in.

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Possible Values | Description |
| max\_depth | integer | *< -1*  *-1*  *0*  *> 0* | Defines the maximum depth of file system traversal when the recurse\_direction behavior is set to a value other than *'none'*.  *< -1*: not permitted.  *-1***:** traverse the file system with no limitation.  *0***:** do not traverse the file system.  *> 0***:** traverse the file system for the specified number of levels.  **Default Value: -1** |
| recurse | string | *‘symlinks’*  *‘directories’*  *‘symlinks and directories’* | Defines how to recurse into the path entity, i.e. what to follow during recursion. Options include symlinks, directories, or both. A max-depth other than 0 MUST be specified for recursion to take place.  *‘symlinks’*: Traverse via only symlinks.  *‘directories’*: Traverse via only directories.  *‘symlinks and directories’*: Traverse via both symlinks and directories. |
| recurse\_direction | string | *'none'*  'up'  *'down'* | Defines the direction to recursively visit the directories on the file system. This behavior only applies with the equality operation on the path entity.  *'none'*: do not traverse the file system.  'up':traverse the file system by recursively visiting the parent directories.  *'down'*:traverse the file system by recursively visiting the child directories.  An error MUST NOT be reported when the max\_depth behavior specifies a certain level of traversal and that level does not exist.  **Default Value: none** |
| recurse\_file\_system | string | *'all'*  *'local'*  *'defined'* | Defines the file system limitation of any searching. This applies to all operations as specified in the path or filepath entity.  In most cases it is recommended that the value of *‘local’* be used to ensure that file system searching is limited to only the local file systems, as searching ‘all’ file systems may have performance implications.  *'all'*:traverse both local and remote file systems.  *'local'*:only traverse the local file systems.  *'defined'*:only traverse the specified file system.  The value of *'defined'* MUST only be used in conjunction with the equality operation because the path or filepath entity must explicitly define a file system.  **Default Value: all** |
| windows\_view | string | *'32\_bit'*  *'64\_bit'* | 64-bit versions of Windows provide an alternate file system view to 32-bit applications[[11]](#footnote-11). This behavior defines which view should be examined.  *'32\_bit'*:check the 32\_bit view of the file system.  *'64\_bit'*:check the 64\_bit view of the file system.  This behavior only applies to 64-bit versions of Windows and MUST NOT be applied on other platforms.  **Default Value: 64-bit** |

## ind-def:textfilecontent\_state

The textfilecontent\_state construct is used by a textfilecontent\_test to specify the information associated with text files that should be checked on various file systems. It is STRONGLY RECOMMENDED that **binary files** are NOT used, as they can contain characters that are detrimental for parsing and XML compatibility.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Type | Multiplicity | Nillable | Description |
| path | oval-def:EntityStateStringType | 0..1 | false | The directory component of the absolute path to a file on the machine. |
| filename | oval-def:EntityStateStringType | 0..1 | false | The name of the file. |
| line | oval-def:EntityStateStringType | 0..1 | false | The line element represents a line in the file and is represented using a regular expression. A single subexpression can be called out using parentheses. Note that when using regular expressions, OVAL supports a common subset of the regular expression character classes, operations, expressions and other lexical tokens defined within Perl 5's regular expression specification[[12]](#footnote-12). |
| subexpression | oval-def:  EntityStateAnySimpleType | 0..1 | false | Each subexpression in the regular expression of the line element is then tested against the value specified in the subexpression element. |
| windows\_view | ind-def:  EntityStateWindowsViewType | 0..1 | false | The windows view value to which this was targeted. This is used to indicate which view (32-bit or 64-bit), the associated State applies to. This entity only applies to 64-bit Microsoft Windows operating systems. |

## ind-sc:textfilecontent\_item

The textfilecontent\_item construct defines the information associated with text files on various file systems.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Type | Multiplicity | Nillable | Description |
| path | oval-sc:EntityItemStringType | 0..1 | false | The name associated with a service. This name is usually the filename of the script file located in the /etc/init.d directory. |
| filename | oval-sc:EntityItemStringType | 0..1 | false | The system runlevel to evaluate. A runlevel is defined as a software configuration of the system that allows only a selected group of processes to exist. |
| line | oval-sc:EntityItemStringType | 0..1 | false | A process is scheduled to be spawned at the specified runlevel. |
| subexpression | oval-sc:  EntityItemAnySimpleType | 0..1 | false | Each subexpression in the regular expression of the line element is then tested against the value specified in the subexpression element. |
| windows\_view | ind-sc:  EntityItemWindowsViewType | 0..1 | false | The windows view value to which this was targeted. This is used to indicate which view (32-bit or 64-bit), the associated State applies to. This entity only applies to 64-bit Microsoft Windows operating systems. |

## ind-def:EntityStateWindowsViewType

The EntityStateWindowsViewType defines the enumeration of possible views associated with 64-bit Microsoft Windows platforms.

|  |  |
| --- | --- |
| Enumeration Value | Description |
| 32\_bit | This value indicates the 32-bit view. |
| 64\_bit | This value indicates the 64-bit view. |
| *<empty string>* | This value indicates that no value has been specified and is permitted here to allow for an empty entity which is associated with a reference to an OVAL Variable. |

## ind-sc:EntityItemWindowsViewType

The EntityItemWindowsViewType defines the enumeration of possible views associated with 64-bit Microsoft Windows platforms.

|  |  |
| --- | --- |
| Enumeration Value | Description |
| 32\_bit | This value indicates the 32-bit view. |
| 64\_bit | This value indicates the 64-bit view. |
| *<empty string>* | This value indicates that no value has been specified and is permitted here to allow for an empty entity which is associated with error and not collected conditions. |

## ind-def:textfilecontent54\_test

The textfilecontent54\_test is used to make assertions on systems about the contents of text files by looking at individual lines. The textfilecontent54\_test MUST reference one textfilecontent54\_object and zero or more textfilecontent54\_states.



### Known Supported Platforms

## ind-def:textfilecontent54\_object

The textfilecontent54\_object construct defines the set of text files whose associated information should be collected and represented as textfilecontent54\_items. The textfilecontent54\_object will only collect regular files[[13]](#footnote-13) on UNIX systems and FILE\_TYPE\_DISK[[14]](#footnote-14) files on Windows systems. It is STRONGLY RECOMMENDED that **binary files** are NOT used, as they can contain characters that are detrimental for parsing and XML compatibility. Also, the *max\_depth* and *recurse\_direction* attributes of the behaviors element do NOT apply to the *filepath* element, only to the *path* and *filename* elements.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Type | Multiplicity | Nillable | Description |
| set | oval-def:set | 0..1 | false | Enables the expression of complex textfilecontent54\_objects that are the result of logically combining and filtering the textfilecontent54\_items that are identified by one or more textfilecontent54\_objects.  Please see the OVAL Language Specification for additional information. |
| behaviors | ind-def:  Textfilecontent54Behaviors | 0..1 | false | Specifies the behaviors that direct how the textfilecontent54\_object collects textfilecontent54\_items from the system. |
| filepath | oval-def:  EntityObjectStringType | 0..1 | false | Specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath. |
| path | oval-def:  EntityObjectStringType | 0..1 | false | The directory component of the absolute path to a file on the machine. |
| filename | oval-def:  EntityObjectStringType | 0..1 | false | The name of the file. |
| pattern | oval-def:  EntityObjectStringType | 0..1 | false | Defines a chunk of text in a file and represented using a regular expression. A subexpression (using parentheses) can call out a piece of the text block to test.  For example, the pattern abc(.\*)xyz would look for a block of text in the file that starts with abc and ends with xyz, with the subexpression being all the characters that exist in between.  The value of the subexpression can then be tested using the subexpression entity of a textfilecontent54\_state. Note that if the pattern, starting at the same point in the file, matches more than one block of text, then it matches the longest.  For example, given a file with abcdefxyzxyzabc, then the pattern abc(.\*)xyz would match the block abcdefxyzxyz. Subexpressions also match the longest possible substrings, subject to the constraint that the whole match be as long as possible, with subexpressions starting earlier in the pattern taking priority over ones starting later.  Note that when using regular expressions, OVAL supports a common subset of the regular expression character classes, operations, expressions and other lexical tokens defined within Perl 5's regular expression specification[[15]](#footnote-15). |
| instance | oval-def:  EntityObjectIntType | 0..1 | false | The instance entity calls out a specific match of the pattern, in which the the first match is assigned to 1, the second to 2, and so on.  The main purpose of this property is to provide uniqueness for different textfilecontent54\_items as a result of multiple matches of a given pattern against the same file. |
| filter | oval-def:filter | 0..\* | false | Allows for the explicit inclusion or exclusion of textfilecontent54\_items from the set of textfilecontent54\_items collected by a textfilecontent54\_object.  Please see the OVAL Language Specification [2] for additional information. |



## ind-def:Textfilecontent54Behaviors

The Textfilecontent54Behaviors construct defines the behaviors that direct how text files are collected from the system. Because Textfilecontent54Behaviors extends the ind-def:FileBehaviors construct, it includes behaviors defined by the construct (*max\_depth, recurse, recurse\_direction, recurse\_file\_system, and windows\_view*, as noted in the diagram above)[[16]](#footnote-16).

Note that using these behaviors may result in some unique results. For example, a double negative type condition might be created where an object entity says include everything except a specific item, but a behavior is used that might then add that item back in.

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Possible Values | Description |
| ignore\_case | boolean | *'true'*  *'false'* | Indicates whether case should be considered when matching system values against the regular expression provided by the pattern entity. This behavior is intended to align with the Perl regular expression 'i' modifier.  *true*: Case will be IGNORED.  *false***:** Case will NOT be IGNORED.  **Default Value: false** |
| multiline | boolean | *'true'*  *'false'* | Enables multiple line semantics in the regular expression provided by the pattern entity. This behavior is intended to align with the Perl regular expression 'm' modifier:  *‘true'*: The'^' and '$' metacharacters will match both at the beginning/end of a string, and immediately after/before newline characters.  *‘false'*: The'^' and '$' metacharacters will match only at the beginning/end of a string.  **Default value: true** |
| singleline | boolean | *'true'*  *'false'* | Enables single line semantics in the regular expression provided by the pattern property.  *'true'*: The '.' metacharacter WILL MATCH newlines.  'false':The '.' metacharacter WILL NOT MATCH newlines.  **Default Value: false** |

## ind-def:textfilecontent54\_state

The textfilecontent54\_state construct is used by a textfilecontent54\_test to specify the information associated with text files that should be checked on various file systems. It is STRONGLY RECOMMENDED that **binary files** are NOT used, as they can contain characters that are detrimental for parsing and XML compatibility.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Type | Multiplicity | Nillable | Description |
| filepath | oval-def:  EntityStateStringType | 0..1 | false | Specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath. |
| path | oval-def:  EntityStateStringType | 0..1 | false | The directory component of the absolute path to a file on the machine. |
| filename | oval-def:  EntityStateStringType | 0..1 | false | The name of the file. |
| pattern | oval-def:  EntityStateStringType | 0..1 | false | Defines a chunk of text in a file and represented using a regular expression. A subexpression (using parentheses) can call out a piece of the text block to test.  Note that when using regular expressions, OVAL supports a common subset of the regular expression character classes, operations, expressions and other lexical tokens defined within Perl 5's regular expression specification[[17]](#footnote-17). |
| instance | oval-def:  EntityStateIntType | 0..1 | false | The instance entity calls out a specific match of the pattern, in which the the first match is assigned to 1, the second to 2, and so on.  The main purpose of this property is to provide uniqueness for different textfilecontent54\_items as a result of multiple matches of a given pattern against the same file. |
| text | oval-def:  EntityStateAnySimpleType | 0..1 | false | The block of text that matched the specified pattern. |
| subexpression | oval-def:  EntityStateAnySimpleType | 0..1 | false | Each subexpression in the regular expression of the line element is then tested against the value specified in the subexpression element. |
| windows\_view | ind-def:  EntityStateWindowsViewType | 0..1 | false | The windows view value to which this was targeted. This is used to indicate which view (32-bit or 64-bit), the associated State applies to. This entity only applies to 64-bit Microsoft Windows operating systems. |

## ind-sc:textfilecontent54\_item

The textfilecontent54\_item construct defines the information associated with text files on various file systems.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Type | Multiplicity | Nillable | Description |
| filepath | oval-sc:  EntityItemStringType | 0..1 | false | Specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath. |
| path | oval-sc:  EntityItemStringType | 0..1 | false | The directory component of the absolute path to a file on the machine. |
| filename | oval-sc:  EntityItemStringType | 0..1 | false | The name of the file. |
| pattern | oval-sc:  EntityItemStringType | 0..1 | false | Defines a chunk of text in a file and represented using a regular expression. A subexpression (using parentheses) can call out a piece of the text block to test.  Note that when using regular expressions, OVAL supports a common subset of the regular expression character classes, operations, expressions and other lexical tokens defined within Perl 5's regular expression specification[[18]](#footnote-18). |
| instance | oval-sc:  EntityItemIntType | 0..1 | false | The instance entity calls out a specific match of the pattern, in which the the first match is assigned to 1, the second to 2, and so on.  The main purpose of this property is to provide uniqueness for different textfilecontent54\_items as a result of multiple matches of a given pattern against the same file. |
| text | oval-sc:  EntityItemAnySimpleType | 0..1 | false | The block of text that matched the specified pattern. |
| subexpression | oval-sc:  EntityItemAnySimpleType | 0..1 | false | Each subexpression in the regular expression of the line element is then tested against the value specified in the subexpression element. |
| windows\_view | ind-sc:  EntityItemWindowsViewType | 0..1 | false | The windows view value to which this was targeted. This is used to indicate which view (32-bit or 64-bit), the associated State applies to. This entity only applies to 64-bit Microsoft Windows operating systems. |

## ind-def:EntityStateWindowsViewType

The EntityStateWindowsViewType defines the enumeration of possible views associated with 64-bit Microsoft Windows platforms.

|  |  |
| --- | --- |
| Enumeration Value | Description |
| 32\_bit | This value indicates the 32-bit view. |
| 64\_bit | This value indicates the 64-bit view. |
| *<empty string>* | This value indicates that no value has been specified and is permitted here to allow for an empty entity which is associated with a reference to an OVAL Variable. |

## ind-sc:EntityItemWindowsViewType

The EntityItemWindowsViewType defines the enumeration of possible views associated with 64-bit Microsoft Windows platforms.

|  |  |
| --- | --- |
| Enumeration Value | Description |
| 32\_bit | This value indicates the 32-bit view. |
| 64\_bit | This value indicates the 64-bit view. |
| *<empty string>* | This value indicates that no value has been specified and is permitted here to allow for an empty entity which is associated with error and not collected conditions. |

## ind-def:variable\_test

The variable\_test allows the value of a variable to be compared to a defined value. For example, one might use this test to validate that a variable being passed in from an external source falls within a specified range. The variable\_test MUST reference one variable\_object and zero or more variable\_states.



### Known Supported Platforms

## ind-def:variable\_object

The variable\_object construct defines the set of variable references whose associated information should be collected and represented as variable\_items[[19]](#footnote-19).



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Type | Multiplicity | Nillable | Description |
| set | oval-def:set | 0..1 | false | Enables the expression of complex process\_objects that are the result of logically combining and filtering the process\_items that are identified by one or more process\_objects.  Please see the OVAL Language Specification for additional information. |
| var\_ref | ind-def:  EntityObjectVariableRefType | 0..1 | false | The id of the variable. |
| filter | oval-def:filter | 0..\* | false | Allows for the explicit inclusion or exclusion of process\_items from the set of process\_items collected by a process\_object.  Please see the OVAL Language Specification [2] for additional information. |

## ind-def:variable\_state

The variable\_state construct is used by a variable\_test to specify indormation about variables and the values associated with them.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Type | Multiplicity | Nillable | Description |
| var\_ref | ind-def: EntityStateVariableRefType | 0..1 | false | The id of the variable. |
| value | oval-def:  EntityStateAnySimpleType | 0..1 | false | The value of the variable. |

## ind-sc:variable\_item

The variable\_item construct defines the information associated with variables.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Type | Multiplicity | Nillable | Description |
| var\_ref | ind-sc: EntityItemVariableRefType | 0..1 | false | The id of the variable. |
| value | oval-sc:  EntityItemAnySimpleType | 0..1 | false | The value of the variable. |

## ind-def:EntityObjectVariableRefType

The EntityObjectVariableRefType defines a string object that has a valid OVAL variable id as the value. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the specified pattern restriction.

|  |  |  |
| --- | --- | --- |
| Datatype Restriction | Additional Restrictions | Explanation |
| oval-def:  EntityObjectStringType | (oval:[A-Za-z0-9\_\-\.]+:var:[1-9][0-9]\*){0,} | Strings with this datatype must be in the format oval:X:var:Y where X is an alphanumeric string (with underscore (\_), dash (-), and dot (.) allowed) and Y is numeric. |
| *<empty string>* | N/A | This value indicates that no value has been specified and is permitted here to allow for an empty entity which is associated with a reference to an OVAL Variable. |

## win-def:EntityStateVariableRefType

The EntityStateVariableRefType defines a string object that has a valid OVAL variable id as the value. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the specified pattern restriction.

|  |  |  |
| --- | --- | --- |
| Datatype Restriction | Additional Restrictions | Explanation |
| oval-def:EntityStateStringType | (oval:[A-Za-z0-9\_\-\.]+:var:[1-9][0-9]\*){0,} | Strings with this datatype must be in the format oval:X:var:Y where X is an alphanumeric string (with underscore (\_), dash (-), and dot (.) allowed) and Y is numeric. |
| *<empty string>* | N/A | This value indicates that no value has been specified and is permitted here to allow for an empty entity which is associated with a reference to an OVAL Variable. |

## win-sc:EntityItemVariableRefType

The EntityObjectGUIDType defines a string object that has a valid OVAL variable id as the value. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the specified pattern restriction.

|  |  |  |
| --- | --- | --- |
| Datatype Restriction | Additional Restrictions | Explanation |
| oval-sc:EntityItemStringType | (oval:[A-Za-z0-9\_\-\.]+:var:[1-9][0-9]\*){0,} | Strings with this datatype must be in the format oval:X:var:Y where X is an alphanumeric string (with underscore (\_), dash (-), and dot (.) allowed) and Y is numeric. |
| *<empty string>* | N/A | This value indicates that no value has been specified and is permitted here to allow for an empty entity which is associated with a reference to an OVAL Variable. |

## ind-def:xmlfilecontent\_test

The xmlfilecontent\_test is used to make assertions about the contents of an xml file. This test allows specific pieces of an xml document specified using xpath to be tested. The xmlfilecontent\_test MUST reference one xmlfilecontent\_object and zero or more xmlfilecontent\_states.



### Known Supported Platforms

## ind-def:xmlfilecontent\_object

The xmlfilecontent\_object construct defines the set of XML file content whose associated information should be collected and represented as xmlfilecontent\_items.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Type | Multiplicity | Nillable | Description |
| set | oval-def:set | 0..1 | false | Enables the expression of complex xmlfilecontent\_objects that are the result of logically combining and filtering the xmlfilecontent\_items that are identified by one or more xmlfilecontent\_objects.  Please see the OVAL Language Specification for additional information. |
| behaviors | ind-def:  FileBehaviors | 0..1 | false | Specifies the behaviors that direct how the xmlfilecontent\_object collects xmlfilecontent\_items from the system. |
| filepath | oval-def:  EntityObjectStringType | 0..1 | false | Specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath. |
| path | oval-def:  EntityObjectStringType | 0..1 | false | Specifies the directory component of the absolute path to a file on the machine. |
| filename | oval-def:  EntityObjectStringType | 0..1 | false | Specifies the name of a file. |
| xpath | oval-def:  EntityObjectStringType | 0..1 | false | Specifies an Xpath expression describing the text node(s) or attribute(s) to look at.  Any valid Xpath 1.0 statement is usable with one exception, at most one field may be identified in the Xpath. This is because the value\_of element in the data section is only designed to work against a single field.  The only valid operator for xpath is  equals since there is an infinite number of possible xpaths and determinining all those that do not equal a give xpath would be impossible. |
| filter | oval-def:filter | 0..\* | false | Allows for the explicit inclusion or exclusion of xmlfilecontent\_items from the set of xmlfilecontent\_items collected by an xmlfilecontent\_object.  Please see the OVAL Language Specification [2] for additional information. |

## ind-def:FileBehaviors

The FileBehaviors construct defines the behaviors that direct how files are collected from the system. Note that using these behaviors may result in some unique results. For example, a double negative type condition might be created where an object entity says include everything except a specific item, but a behavior is used that might then add that item back in.

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Possible Values | Description |
| max\_depth | integer | *< -1*  *-1*  *0*  *> 0* | Defines the maximum depth of file system traversal when the recurse\_direction behavior is set to a value other than *'none'*.  *< -1*: not permitted.  *-1***:** traverse the file system with no limitation.  *0***:** do not traverse the file system.  *> 0***:** traverse the file system for the specified number of levels.  **Default Value: -1** |
| recurse | string | *‘symlinks’*  *‘directories’*  *‘symlinks and directories’* | Defines how to recurse into the path entity, i.e. what to follow during recursion. Options include symlinks, directories, or both. A max-depth other than 0 MUST be specified for recursion to take place.  *‘symlinks’*: Traverse via only symlinks.  *‘directories’*: Traverse via only directories.  *‘symlinks and directories’*: Traverse via both symlinks and directories. |
| recurse\_direction | string | *'none'*  'up'  *'down'* | Defines the direction to recursively visit the directories on the file system. This behavior only applies with the equality operation on the path entity.  *'none'*: do not traverse the file system.  'up':traverse the file system by recursively visiting the parent directories.  *'down'*:traverse the file system by recursively visiting the child directories.  An error MUST NOT be reported when the max\_depth behavior specifies a certain level of traversal and that level does not exist.  **Default Value: none** |
| recurse\_file\_system | string | *'all'*  *'local'*  *'defined'* | Defines the file system limitation of any searching. This applies to all operations as specified in the path or filepath entity.  In most cases it is recommended that the value of *‘local’* be used to ensure that file system searching is limited to only the local file systems, as searching ‘all’ file systems may have performance implications.  *'all'*:traverse both local and remote file systems.  *'local'*:only traverse the local file systems.  *'defined'*:only traverse the specified file system.  The value of *'defined'* MUST only be used in conjunction with the equality operation because the path or filepath entity must explicitly define a file system.  **Default Value: all** |
| windows\_view | string | *'32\_bit'*  *'64\_bit'* | 64-bit versions of Windows provide an alternate file system view to 32-bit applications[[20]](#footnote-20). This behavior defines which view should be examined.  *'32\_bit'*:check the 32\_bit view of the file system.  *'64\_bit'*:check the 64\_bit view of the file system.  This behavior only applies to 64-bit versions of Windows and MUST NOT be applied on other platforms.  **Default Value: 64-bit** |

## ind-def:xmlfilecontent\_state

The xmlfilecontent\_state construct is used by a xmlfilecontent\_test to specify the XML file content to be collected.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Type | Multiplicity | Nillable | Description |
| filepath | oval-def:EntityStateStringType | 0..1 | false | Specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath. |
| path | oval-def:EntityStateStringType | 0..1 | false | Specifies the directory component of the absolute path to a file on the machine. |
| filename | oval-def:EntityStateStringType | 0..1 | false | Specifies the name of a file. |
| xpath | oval-def:EntityStateStringType | 0..1 | false | Specifies an Xpath expression describing the text node(s) or attribute(s) to look at.  Any valid Xpath 1.0 statement is usable with one exception, at most one field may be identified in the Xpath. This is because the value\_of element in the data section is only designed to work against a single field.  The only valid operator for xpath is  equals since there is an infinite number of possible xpaths and determinining all those that do not equal a give xpath would be impossible. |
| value\_of | oval-def:EntityStateIntType | 0..1 | false | Checks the value(s) of the text node(s) or attribute(s) found. |
| windows\_view | ind-def:  EntityStateWindowsViewType | 0..1 | false | The windows view value to which this was targeted. This is used to indicate which view (32-bit or 64-bit), the associated State applies to. This entity only applies to 64-bit Microsoft Windows operating systems. |

## ind-sc:xmlfilecontent\_item

The xmlfilecontent\_item construct defines the XML file content to be collected.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Type | Multiplicity | Nillable | Description |
| filepath | oval-sc:EntityItemStringType | 0..1 | false | Specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath. |
| path | oval-sc:EntityItemStringType | 0..1 | false | Specifies the directory component of the absolute path to a file on the machine. |
| filename | oval-sc:EntityItemStringType | 0..1 | false | Specifies the name of a file. |
| xpath | oval-sc:EntityItemStringType | 0..1 | false | Specifies an Xpath expression describing the text node(s) or attribute(s) to look at.  Any valid Xpath 1.0 statement is usable with one exception, at most one field may be identified in the Xpath. This is because the value\_of element in the data section is only designed to work against a single field.  The only valid operator for xpath is  equals since there is an infinite number of possible xpaths and determinining all those that do not equal a give xpath would be impossible. |
| value\_of | oval-sc:EntityItemIntType | 0..1 | false | Checks the value(s) of the text node(s) or attribute(s) found. |
| windows\_view | ind-sc:  EntityItemWindowsViewType | 0..1 | false | The windows view value to which this was targeted. This is used to indicate which view (32-bit or 64-bit), the associated State applies to. This entity only applies to 64-bit Microsoft Windows operating systems. |

# Appendix A – Normative References

[1] RFC 2119 – Key words for use in RFCs to Indicate Requirement Levels

<http://www.ietf.org/rfc/rfc2119.txt>

[2] The OVAL Language Specification

<http://oval.mitre.org/language/version5.10#specification>

# Appendix B - Change Log

# 

# Appendix C – Terms and Acronyms

1. For more information see <https://oval.mitre.org/about/termsofuse.html> [↑](#footnote-ref-1)
2. For more information see <https://oval.mitre.org/> [↑](#footnote-ref-2)
3. For more information see <http://en.wikipedia.org/wiki/Namespace_(computer_science)> [↑](#footnote-ref-3)
4. For more information about environment variables in the various OSes, see <http://environmentvariables.org/> [↑](#footnote-ref-4)
5. For more information about environment variables in the various OSes, see <http://environmentvariables.org/> [↑](#footnote-ref-5)
6. For more information about environment variables in the various OSes, see <http://environmentvariables.org/> [↑](#footnote-ref-6)
7. For more information about environment variables in the various OSes, see <http://environmentvariables.org/> [↑](#footnote-ref-7)
8. See <http://tldp.org/LDP/intro-linux/html/sect_03_01.html> for more information about regular files. Virtually every file that contains normal data is called a *regular file*. Directories, special files, links, sockets, and named pipes are NOT regular files. Use ls –l and look for a dash (-) before the file permissions to determine this. [↑](#footnote-ref-8)
9. This can be determined via the GetFileType function in Windows. See <http://msdn.microsoft.com/en-us/library/windows/desktop/aa364960(v=vs.85).aspx> for more information. Virtually any file with normal data is considered a disk file, except for a character file referring to an LPT device or console (FILE\_TYPE\_CHAR), a socket or pipe (FILE\_TYPE\_PIPE), or one that is considered unknown assuming the GetLastError function returned NO\_ERROR (FILE\_TYPE\_UNKNOWN). [↑](#footnote-ref-9)
10. For more information on the supported regular expression syntax in OVAL see: <http://oval.mitre.org/language/about/re_support_5.6.html>. [↑](#footnote-ref-10)
11. For more information see <http://msdn.microsoft.com/en-us/library/aa384187(v=vs.85).aspx> [↑](#footnote-ref-11)
12. For more information on the supported regular expression syntax in OVAL see: <http://oval.mitre.org/language/about/re_support_5.6.html>. [↑](#footnote-ref-12)
13. See <http://tldp.org/LDP/intro-linux/html/sect_03_01.html> for more information about regular files. Virtually every file that contains normal data is called a *regular file*. Directories, special files, links, sockets, and named pipes are NOT regular files. Use ls –l and look for a dash (-) before the file permissions to determine this. [↑](#footnote-ref-13)
14. This can be determined via the GetFileType function in Windows. See <http://msdn.microsoft.com/en-us/library/windows/desktop/aa364960(v=vs.85).aspx> for more information. Virtually any file with normal data is considered a disk file, except for a character file referring to an LPT device or console (FILE\_TYPE\_CHAR), a socket or pipe (FILE\_TYPE\_PIPE), or one that is considered unknown assuming the GetLastError function returned NO\_ERROR (FILE\_TYPE\_UNKNOWN). [↑](#footnote-ref-14)
15. For more information on the supported regular expression syntax in OVAL see: <http://oval.mitre.org/language/about/re_support_5.6.html>. [↑](#footnote-ref-15)
16. For more information about ind-def:FileBehaviors, see Section 2.6. [↑](#footnote-ref-16)
17. For more information on the supported regular expression syntax in OVAL see: <http://oval.mitre.org/language/about/re_support_5.6.html>. [↑](#footnote-ref-17)
18. For more information on the supported regular expression syntax in OVAL see: <http://oval.mitre.org/language/about/re_support_5.6.html>. [↑](#footnote-ref-18)
19. For more information see <http://unixhelp.ed.ac.uk/CGI/man-cgi?ps> [↑](#footnote-ref-19)
20. For more information see <http://msdn.microsoft.com/en-us/library/aa384187(v=vs.85).aspx> [↑](#footnote-ref-20)