

# - Open Vulnerability and Assessment Language - Element Dictionary

- Schema: Core Results
- Version: 5.0 release candidate 3
- Release Date: 26 May 2006

The following is a description of the elements, types, and attributes that compose the core schema for encoding Open Vulnerability and Assessment Language (OVAL) Results. Each of the elements, types, and attributes that make up the Core Results Schema are described in detail and should provide the information necessary to understand what each object represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between these objects is not outlined here.

The OVAL Schema is maintained by The MITRE Corporation and developed by the public OVAL Community. For more information, including how to get involved in the project and how to submit change requests, please visit the OVAL website at <http://oval.mitre.org>.

---

---

## < oval\_results >

The oval\_results element is the root of an OVAL Results Document. Its purpose is to bind together the four major sections of a results file - generator, directives, oval\_definitions, and results - which are the children of the root element. It must contain exactly one generator section, one directives section, and one results section. The required generator section provides information about when the results file was compiled and under what version. The required directives section presents flags describing what information has been included in the results file. The oval\_definitions section is optional and dependent on the directives that are set. Its purpose is to provide an exact copy of the definitions evaluated for the results file. The required results section holds all the results of the evaluated definitions. The optional Signature element allows an XML Signature as defined by the W3C to be attached to the document. This allows authentication and data integrity to be provided to the user. Enveloped signatures are supported.

Child Elements	Type	MinOccurs	MaxOccurs
generator	oval:GeneratorType	1	1
directives	<a href="#">oval-res:DirectivesType</a>	1	1
oval-def:oval_definitions	n/a	0	1
results	<a href="#">oval-res:ResultsType</a>	1	1
ds:Signature	n/a	0	1

---

---

## == DirectivesType ==

The DirectivesType complex type presents flags describing what information has been included in the results file. There are six possible results (true, false, unknown, error, not evaluated, and not applicable) for an evaluation of an OVAL Definition. The directives state which of these results are being reported in the results file. For example, a results file dealing with vulnerabilities might only present to the user the definitions that returned a true result meaning the vulnerability exists. A different result file dealing with compliance definitions might want to report the results of all definitions except those not evaluated.

Child Elements	Type	MinOccurs	MaxOccurs
definition_true	<a href="#">oval-res:DirectiveType</a>	1	1
definition_false	<a href="#">oval-res:DirectiveType</a>	1	1
definition_unknown	<a href="#">oval-res:DirectiveType</a>	1	1
definition_error	<a href="#">oval-res:DirectiveType</a>	1	1
definition_not_evaluated	<a href="#">oval-res:DirectiveType</a>	1	1
definition_not_applicable	<a href="#">oval-res:DirectiveType</a>	1	1

## == DirectiveType ==

Each directive determines whether or not certain results are included in the results file. The required attribute controls this by providing a true or false for the specific directive. The optional content attribute controls how much information about the specific result is provided. For example, thin content would only be the id of the definition and the result, while a full content set would be the definition id with the result along with all the test ids and their results. Please refer to the contentEnumeration for details about the different content options.

### Attributes:

- 
- |            |   |                              |
|------------|---|------------------------------|
| - reported | xsd:boolean                                 | (required)                   |
| - content  | <a href="#">oval-res:ContentEnumeration</a> | (optional -- default='full') |
- 
- 

## == ResultsType ==

The ResultsType complex type is a container for one or more system elements. Each system element defines the results associated with an individual system. Please refer to the description of SystemType for more information about an individual system element.

Child Elements	Type	MinOccurs	MaxOccurs
system	<a href="#">oval-res:SystemType</a>	1	unbounded

## == SystemType ==

The SystemType complex type holds the evaluation results of the definitions and tests, as well as a copy of the OVAL System Characteristics used to perform the evaluation. The definitions section holds the results of the definitions and the tests section holds the results of the tests. The oval\_system\_characteristics section is a copy of the system characteristics file used to perform the evaluation of the OVAL Definitions. Note that the oval\_definitions part of the system characteristics file should be left out as the definition information has already been included at the top of the results file.

Child Elements	Type	MinOccurs	MaxOccurs
definitions	<a href="#">oval-res:DefinitionsType</a>	0	1
tests	<a href="#">oval-res:TestsType</a>	0	1
oval-sc:oval_system_characteristics	n/a	1	1

---

---

## == DefinitionsType ==

The DefinitionsType complex type is a container for one or more definition elements. Each definition element holds the result of the evaluation of an OVAL Definition. Please refer to the description of DefinitionType for more information about an individual definition element.

Child Elements	Type	MinOccurs	MaxOccurs
definition	<a href="#">oval-res:DefinitionType</a>	1	unbounded

## == DefinitionType ==

The DefinitionType complex type holds the result of the evaluation of an OVAL Definition. The message element holds an error message or some other string that the analysis engine wishes to pass along. In addition, the optional criteria element provides the results of the individual pieces of the criteria. Please refer to the description of the CriteriaType for more information.

The required definition\_id attribute is the OVAL id of the definition. The required version attribute is the specific version of the OVAL Definition used during analysis. The optional variable\_instance attribute is a unique id that differentiates every unique instance (based on the value of variables used) of a definition in the OVAL Results file. Languages that include OVAL might reference the same definition multiple times. Each time a different set of values is supplied for the variables, a new instance of the definition result must be created. (definitions that do not use variables can only have one unique instance) The inclusion of a unique instance identifier will allow the OVAL results file to report the correct result of a definition for each combination of supplied values. The required result attribute holds the result of the evaluation. Please refer to the description of the resultEnumeration for details about the different result values.

**Attributes:**

- 
- definition\_id      oval:DefinitionIDPattern      (required)
  - version            xsd:integer                              (required)
  - variable\_instance   xsd:integer                              (optional -- default='1')
  - result              [oval-res:ResultEnumeration](#)      (required)

Child Elements	Type	MinOccurs	MaxOccurs
message	oval:MessageType	0	unbounded
criteria	<a href="#">oval-res:CriteriaType</a>	0	1

**== CriteriaType ==**

The CriteriaType complex type describes the high level container for all the tests and represents the meat of the definition. Each criteria can contain other criteria elements in a recursive structure allowing complex logical trees to be constructed. Each referenced test is represented by a criterion element. Please refer to the description of the CriterionType for more information about an individual criterion element. The optional extend\_definition element allows existing definitions to be included in the criteria. Refer to the description of the ExtendDefinitionType for more information.

The required operator attribute provides the logical operator that binds the different statements inside a criteria together. The optional negate attribute signifies that the result of an extended definition should be negated during analysis. For example, consider a definition that evaluates TRUE if a certain software is installed. By negating the definition, it now evaluates to TRUE if the software is NOT installed. The required result attribute holds the result of the evaluation of the criteria. Note that this would be after any negation operation has been applied. Please refer to the description of the resultEnumeration for details about the different result values.

**Attributes:**

- 
- operator      oval:OperatorEnumeration      (required)
  - negate        xsd:boolean                              (optional -- default='false')
  - result        [oval-res:ResultEnumeration](#)      (required)

Child Elements	Type	MinOccurs	MaxOccurs
criteria	<a href="#">oval-res:CriteriaType</a>		
criterion	<a href="#">oval-res:CriterionType</a>		
extend_definition	<a href="#">oval-res:ExtendDefinitionType</a>		

**== CriterionType ==**

The CriterionType complex type identifies a specific test that is included in the definition's criteria.

The required test\_id attribute is the actual id of the included test. The required version attribute is the specific version of the OVAL Test used during analysis. The optional variable\_instance attribute

differentiates between unique instances of a test. This can happen when a test includes a variable reference and different values for that variable are used by different definitions. The optional negate attribute signifies that the result of an individual test should be negated during analysis. For example, consider a test that evaluates to TRUE if a specific patch is installed. By negating this test, it now evaluates to TRUE if the patch is NOT installed. The required result attribute holds the result of the evaluation. Please refer to the description of the resultEnumeration for details about the different result values.

**Attributes:**

---

- test_ref	oval:TestIDPattern	(required)
- version	xsd:integer	(required)
- variable_instance	xsd:integer	(optional -- default='1')
- negate	xsd:boolean	(optional -- default='false')
- result	<a href="#">oval-res:ResultEnumeration</a>	(required)

**== ExtendDefinitionType ==**

The ExtendDefinitionType complex type identifies a specific definition that has been extended by the criteria.

The required definition\_id attribute is the actual id of the extended definition. The required version attribute is the specific version of the OVAL Definition used during analysis. The optional variable\_instance identifier is a unique id that differentiates every unique instance of a definition in the OVAL Results file based on the combination of variable values used. Languages that include OVAL might reference the same definition multiple times. Each time a different set of values is supplied for the variables, a new instance of the definition must be created. (definitions that do not use variables can only have one unique instance) The inclusion of a unique instance identifier will allow the OVAL results file to report the correct result of a definition for each combination of supplied values. The optional negate attribute signifies that the result of an extended definition should be negated during analysis. For example, consider a definition that evaluates TRUE if a certain software is installed. By negating the definition, it now evaluates to TRUE if the software is NOT installed. The required result attribute holds the result of the evaluation. Please refer to the description of the resultEnumeration for details about the different result values.

**Attributes:**

---

- definition_ref	oval:DefinitionIDPattern	(required)
- version	xsd:integer	(required)
- variable_instance	xsd:integer	(optional -- default='1')
- negate	xsd:boolean	(optional -- default='false')
- result	<a href="#">oval-res:ResultEnumeration</a>	(required)

---

**== TestsType ==**

The TestsType complex type is a container for one or more test elements. Each test element holds the

result of the evaluation of an OVAL Test. Please refer to the description of TestType for more information about an individual test element.

Child Elements	Type	MinOccurs	MaxOccurs
test	<a href="#">oval-res:TestType</a>	1	unbounded

## == TestType ==

The TestType complex type provides a reference to every object that matched the object section of the original test. The optional message element holds an error message or some other string that the analysis engine wishes to pass along. The overall result of the test is determined by the results of each matching object and the check attribute. Please refer to the TestedObjectType for more details. The optional tested\_variable elements hold the value of each variable used by the test during evaluation. This includes the values used in both OVAL Objects and OVAL States. If a variable represents an array of values, then multiple tested\_variable elements would exist with the same variable\_id attribute. Please refer to the description of TestedVariableType for more information.

The required test\_id attribute identifies the test, and must conform to the format specified by the testidPattern simple type. The required version attribute is the specific version of the OVAL Test used during analysis. The optional variable\_instance attribute differentiates between unique instances of a test. This can happen when a test includes a variable reference and different values for that variable are used by different definitions. The required check attribute determines what group of objects to test. (For example: Should the test check that all files match a specified version or that at least one file matches the specified version?) The valid check values are explained in the description of the checkEnumeration simple type found in the core definitions schema. The required result attribute holds the result of the evaluation. Please refer to the description of the resultEnumeration for details about the different result values.

### Attributes:

---

- test_id	oval:TestIDPattern	(required)
- version	xsd:integer	(required)
- variable_instance	xsd:integer	(optional -- default='1')
- check	oval:CheckEnumeration	(required)
- result	<a href="#">oval-res:ResultEnumeration</a>	(required)

Child Elements	Type	MinOccurs	MaxOccurs
message	oval:MessageType	0	unbounded
tested_item	<a href="#">oval-res:TestedItemType</a>	0	unbounded
tested_variable	<a href="#">oval-res:TestedVariableType</a>	0	unbounded

## == TestedItemType ==

The TestedItemType complex type holds a reference to each system characteristic item that matches the object specified in a test. Details of the item can be found in the oval\_system\_characteristics section of the

OVAL Results file by using the required `item_id`. The optional message element holds an error message or some other string that the analysis engine wishes to pass along. The required result attribute holds the result of the evaluation of the individual item as it relates to the state specified by the test. Please refer to the description of the `resultEnumeration` for details about the different result values.

**Attributes:**

- `item_id`      `oval:ItemIDPattern`                      (required)
- `result`        [oval-res:ResultEnumeration](#)                      (required)

Child Elements	Type	MinOccurs	MaxOccurs
message	oval:MessageType	0	unbounded

**== TestedVariableType ==**

The `TestedVariableType` complex type holds the value to a variable used during the evaluation of a test. Of special importance are the values of any external variables used since these values are not captured in either the definition or system characteristic files. If a variable is represented by an array of values, then multiple elements of `TestedVariableType`, each with the same `variable_id` attribute, would exist. The required `variable_id` attribute is the unique id of the variable that was used.

**Attributes:**

- `variable_id`      `oval:VariableIDPattern`                      (required)

<b>Simple Content</b>	<code>xsd:anySimpleType</code>
-----------------------	--------------------------------

**-- ContentEnumeration --**

Defines the valid values for the directives controlling the expected content of the results file. The specific content that is expected with each value is defined by a style sheet that complements the OVAL Results Schema. Please refer to these style sheets for more information.

Value	Description
thin	A value of 'thin' means only the minimal amount of information will be provided. This is the id associated with an evaluated OVAL Definition and the result of the evaluation. The criteria child element of a definition should not be present when providing thin results. In addition, system characteristic information for the objects used by the given definition should not be presented.

full	A value of 'full' means that very detailed information will be provided allowing in-depth reports to be generated from the results. In addition to the results of the evaluated definition, the results of all extended definitions and tests included in the criteria as well as the actual information collected off the system will be presented.
------	--

**-- ResultEnumeration --**

Define acceptable result values for the evaluation of an OVAL Definition or an OVAL Test.

Value	Description
true	When evaluating a definition or test, a result value of 'true' means that the characteristics being evaluated match the information represented in the system characteristic file.
false	When evaluating a definition or test, a result value of 'false' means that the characteristics being evaluated do not match the information represented in the system characteristic file.
unknown	When evaluating a definition or test, a result value of 'unknown' means that the characteristics being evaluated can not be found in the system characteristic file. (or the characteristics can be found but collected object flag is 'not collected') For example, assume you have a definition that tests a file, but when you look at the system characteristic file, data pertaining to that file can not be found. The lack of an object (in the collected_object section) for this file in the SC file means that no attempt was made to even try and collect information about the file. So you do not know what the result would be if it was collected.
error	When evaluating a definition or test, a result value of 'error' means that the characteristics being evaluated exist in the system characteristic file but there was an error either collecting information or in performing analysis. For example, xsi:nil might be set on an object entity, but then the entity is compared to a state entity with a value, thus producing an error.
not evaluated	When evaluating a definition or test, a result value of 'not evaluated' means that a choice was made not to evaluate the given definition or test. The actual



	result is in essence unknown since if evaluation had occurred it could have been either true or false.
not applicable	When evaluating a definition or test, a result value of 'not applicable' means that the definition or test being evaluated is not valid on the given platform. For example, trying to collect Linux RPM information on a Windows system.