The OVAL Community Guidelines

The OVAL community with notable contributions by David Ries (jo

Apr 28, 2022

The Guidelines

Welcome to the guidelines for OVAL, the Open Vulnerability and Assessment Language. These guidelines are designed to explain everything you need to know to start contributing to OVAL (or link you to places to ask questions, should the explanations not suffice), as well as provide a variety of standards and resources to the community.

CHAPTER 1

What is OVAL?

OVAL is an open language built by security experts, system administrators, and software developers to universalize assessment and reporting on the state of computer systems.

CHAPTER 2

Who is the OVAL Community?

The OVAL Community is the group responsible for proposals about anything and everything OVAL related. It comprises Members, Area Supervisors, the Leadership Board, and the Sponsor. The community maintains, fixes, and improves OVAL through an established governance process.

CHAPTER 3

Learn More

OVAL Community
Repository Registry
CIS OVAL Repository (GitHub)
CIS OVAL Repository (Static Site)
Mailing Lists

CHAPTER 4

Get Involved

Language Development
Language Proposal Process
OVAL Community GitHub

Content Development
Contributing OVAL Content
CIS OVAL Repository

Discussion Forums
Mailing Lists

CHAPTER 5

License

Terms of Use

5.1 Getting Started

Are you new to OVAL? Wondering what it is and how it's used? Read on!

5.1.1 What is OVAL?

OVAL is an open, standardized assertion language written in XML that standardizes how to assess and report on the machine state of computer systems. Used by the U.S. Government, the Center for Internet Security, Cisco, and McAfee, among many others, it is the most mature and widely adopted open source standard for security assessment. With the goal of easing interoperability between security tools, it includes content for vulnerability assessment, configuration management, system inventory, and patch management. Security experts, system administrators, and software developers from industry, government, and academia have collaborated to write OVAL, and this consensus is one of its greatest attributes.

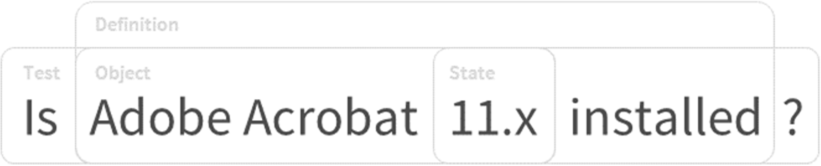
Anyone can write OVAL, and we always welcome new contributors.

5.1.2 OVAL Use Cases

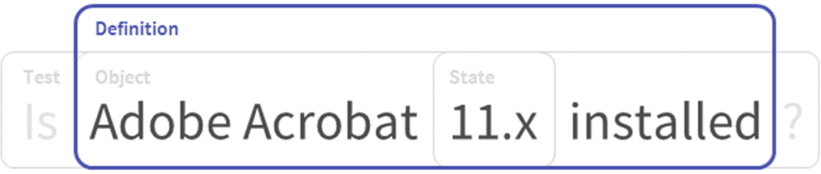
OVAL is primarily used for assessing vulnerabilities in security configurations. OVAL content can also be used in other ways, documented in the Use Cases.

5.1.3 OVAL Structure

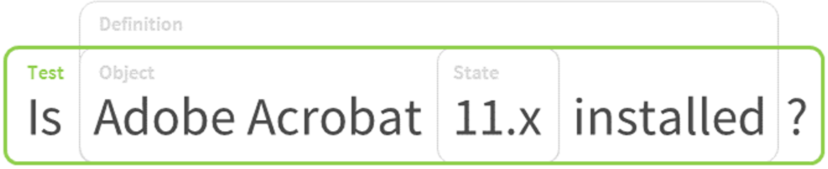
OVAL can be broken down into a series of components that together represent a check, validation, or idea. This can generally be expressed as a prose sentence:



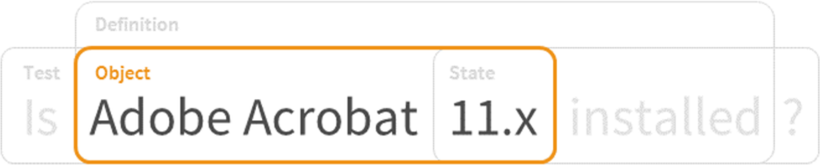
This is expressed as a definition, which references or includes the other components as seen below.



definitions Definitions are specifications of what endpoint information should be checked and what corresponding values are expected to be found, as well as how to interpret the results of that comparison. They comprise one or more tests, which taken together represent an externally meaningful datum, such as a vulnerability state or inventory status.

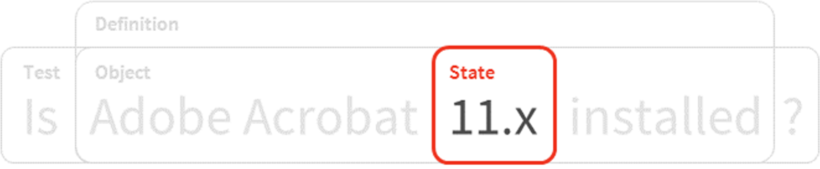


tests Tests are the concrete building blocks of definitions. They specify the relationship between an OVAL Object and zero or more OVAL States, matching the information to be collected with the corresponding values expected to be found.



objects Objects define what should be collected from an endpoint.

> *A concrete OVAL Object may define a set of 0 or more OVAL Behaviors. OVAL Behaviors are actions that can further specify the set of OVAL Items that match an OVAL Object.*



states States are the expected values from an object that are compared to the information collected from an endpoint.

variables Variables provide a way to group one or more values for consistent reference within other OVAL content.

5.1.4 An Annotated Sample

Below is a sample OVAL definition file:

```
<?xml version="1.0" encoding="UTF-8"?>
<oval_definitions xmlns="http://oval.mitre.org/XMLSchema/oval-definitions-5"␣
↪xmlns:oval="http://oval.mitre.org/XMLSchema/oval-common-5" xmlns:xsi="http://www.w3.
↪org/2001/XMLSchema-instance" xsi:schemaLocation="http://oval.mitre.org/XMLSchema/
↪oval-common-5 oval-common-schema.xsd http://oval.mitre.org/XMLSchema/oval-
↪definitions-5 oval-definitions-schema.xsd">
```

(continues on next page)

(continued from previous page)

```
<generator>
<!--
The generator element provides metadata about the tool/application used to develop␣
↪the OVAL Content.
-->
<oval:schema_version>5.11.2</oval:schema_version>
<oval:timestamp>2018-07-31T17:30:20</oval:timestamp>
</generator>

<definitions>
 <!--
 The definitions element contains the OVAL definition(s) to be exchanged.
 -->
 <definition class="compliance" id="oval:org.oval-community.example:def:1" version="1
↪">
 <!--
 This definition checks compliance.
 -->
    <metadata>
       <!--
       The metadata element contains information about the definition, including its␣
↪title and description. This definition checks whether WinRM traffic is encrypted or␣
↪not.
       -->
          <title>WinRM Traffic Must be Encrypted</title>
          <affected family="windows">
             <platform>Microsoft Windows Server 2016</platform>
          </affected>
       <reference ref_id="CCE-46378-6" ref_url="http://cce.mitre.org" source="CCE"/>
       <description>The Windows Remote Management (WinRM) client must not allow␣
↪unencrypted traffic.</description>
    </metadata>
 <notes>
 <note>This sample was based on an OVAL definition included in the Windows Server␣
↪2016 STIG available at https://iase.disa.mil/ </note>
 </notes>
 criteria operator="AND">
    <!--
    The criteria element specifies the assertion to be tested using information␣
↪gathered from the endpoint.
    -->
       <criterion comment="Verifies 'WinRM Client: Allow unencrypted traffic' is set␣
↪to 'Disabled'" test_ref="oval:org.oval-community.example:tst:1"/>\
          <!--
          The criterion elements define logical terms in the assertion. This criteria␣
↪only uses 1 criterion element to check if 'WinRM Client: Allow unencrypted traffic'␣
↪is set to 'Disabled'.

          By default, the truth values returned by the tests are AND'ed to determine␣
↪the truth value of the assertion.
          -->
    </criteria>
 </definition>
</definitions>

<tests>
 <!--
```

(continues on next page)

(continued from previous page)

```
 The tests element contains the OVAL Test(s). OVAL Tests specify what to search for␣
↪on an endpoint (i.e., objects) and what is expected to be found (i.e., states).

 The registry_test is used to check information in the Windows registry.
 -->
    <registry_test check="all" check_existence="at_least_one_exists" comment="WinRM␣
↪Client: Allow unencrypted traffic is set to 'Disabled'" id="oval:org.oval-community.
↪example:tst:1" version="1" xmlns="http://oval.mitre.org/XMLSchema/oval-definitions-5
↪#windows">
       <!--
       This registry_test checks that 'Allow unencrypted traffic' is set to 'Disabled
↪'.
       -->
       <object object_ref="oval:org.oval-community.example:obj:1"/>
       <state state_ref="oval:org.oval-community.example:ste:1"/>
    </registry_test>
</tests>

<objects>
 <!--
 The objects element contains the OVAL Object(s).

 The registry_object is used to search for information in the Windows registry.
 -->
    <registry_object comment="WinRM Cl ient: AllowUnencryptedTraffic registry key" id=
↪"oval:org.oval-community.example:obj:1" version="1" xmlns="http://oval.mitre.org/
↪XMLSchema/oval-definitions-5#windows">
       <!--
       This registry_object specifies that the registry key containing the policy␣
↪definition for 'WinRM Client: Allow unencrypted traffic' should be checked.
       -->
          <hive datatype="string" operation="equals">HKEY_LOCAL_MACHINE</hive>
          <key datatype="string" operation="equals">
↪Software\Policies\Microsoft\Windows\WinRM\Client</key>
          <name datatype="string" operation="equals">AllowUnencryptedTraffic</name>
    </registry_object>
</objects>

<states>
 <!--
 The states element contains the OVAL State(s).

 The registry_state is used to describe information expected to be found in the␣
↪Windows registry.
 -->
    <registry_state comment="Reg_Dword equals 0" id="oval:org.oval-community.
↪example:ste:1" version="1" xmlns="http://oval.mitre.org/XMLSchema/oval-definitions-5
↪#windows">
       <type>reg_dword</type>
          <!--
          This registry_state specifies that an integer matching '0' is expected to␣
↪be found in the registry.
          -->
       <value datatype="int" operation="equals">0</value>
    </registry_state>
</states>
```

(continues on next page)

(continued from previous page)

```
</oval_definitions>
```

5.1.5 OVAL Features

- XML- and assertion-based language
- implementation-neutral, semantic content authoring
- enables enforcement of script-free, read-only policy
- supports content reuse
- complex first order logic
- variables in a variety of functions for string manipulation
- supports technology-neutral policy authoring
- extensible
- supports trust management through digital signatures and verifications
- automatically checkable for conformance with standard
- brings consistency and transparency to the results produced by security scanning tools
- assists in the exchange of machine-readable information between security tools
- reduces the need for IT Security Professionals to learn the proprietary languages of each of their tools

Use OVAL to:

- make implementation-neutral assertions about platforms and their machine states (e.g. files, registry keys, etc.)
- express policy content without defining implementation method

5.1.6 The OVAL Schemas

OVAL comprises a set of schemas, which correspond to unique Models that establish the logical framework for making assertions about the posture of an endpoint. The Models provide the building blocks for representing the expected and actual states of endpoints and the results of the comparison of those elements.

There are two main sets of schemas: Core and Platform Extensions. The Core Schemas form the foundation of the language, while Platform Extensions extend the Core Schemas to support different platforms, such as Windows, Linux, and Cisco IOS.

5.1.7 Related Standards

XCCDF The eXtensible Configuration Checklist Description Format language describes security checklists. Documents in this format may reference OVAL components or documents, as well as ones from other standards, creating a portable and flexible checklist.

SCE The Script Check Engine complements OVAL with scripts that check things that OVAL cannot or does not. SCE results files are created as an XML. By using XLST transformations, OVAL and SCE results can be aggregated into a single HTML file or PDF document.

CPE The Common Platform Enumeration provides a standard naming scheme for IT platforms and systems. OVAL uses it to consistently identify the target platforms of checks and definitions.

Datastreams **Datastream** is a format that consolidates multiple SCAP components into a single file (including OVAL).

ARF, or the **Asset Reporting Format**, is also called Result Datastream. It consolidates multiple results files into one.

5.1.8 Next Steps

- Additional Resources

5.2 OVAL Schema Documentation

This is an index page for quick reference to generated schema documentation.

5.2.1 Core Schemas

- *Common*
- *Definitions*
- *System-Characteristics*
- *Results*

5.2.2 OVAL Interpreter Schemas

- *Directives*
- *Evaluation-IDs*
- *External Variables*

5.2.3 Platform Schemas

Independent

- Platform-Independent: *Definitions*, *System Characteristics*

Mobile Devices

- Apple iOS: *Definitions*, *System Characteristics*
- Google Andriod: *Definitions*, *System Characteristics*

Network Devices

- Cisco ASA: *Definitions*, *System Characteristics*
- Cisco CATOS: *Definitions*, *System Characteristics*
- Cisco IOS: *Definitions*, *System Characteristics*
- Cisco IOS-XE: *Definitions*, *System Characteristics*
- Cisco PIX: *Definitions*, *System Characteristics*
- Juniper JunOS: *Definitions*, *System Characteristics*
- NETCONF: *Definitions*, *System Characteristics*

Microsoft-Specific

- Windows: *Definitions*, *System Characteristics*
- Sharepoint: *Definitions*, *System Characteristics*

Unix Operating Systems

- Unix (Generic): *Definitions*, *System Characteristics*
- Apple MacOS: *Definitions*, *System Characteristics*
- FreeBSD: *Definitions*, *System Characteristics*
- HP-UX: *Definitions*, *System Characteristics*
- IBM AIX: *Definitions*, *System Characteristics*
- Linux: *Definitions*, *System Characteristics*
- Oracle Solaris: *Definitions*, *System Characteristics*
- VMWare ESX: *Definitions*, *System Characteristics*

Application-Specific

- Apache: *Definitions*, *System Characteristics*

Open Vulnerability and Assessment Language: Core Common

- Schema: Core Common
- Version: 5.11.2
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the common types that are shared across the different schemas within Open Vulnerability and Assessment Language (OVAL). Each type is described in detail and should provide the information necessary to understand what each represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between these type is not outlined here.

The OVAL Schema is maintained by the 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.cisecurity.org.

< deprecated_info >

The deprecated_info element is used in documenting deprecation information for items in the OVAL Language. It is declared globally as it can be found in any of the OVAL schemas and is used as part of the appinfo documentation and therefore it is not an element that can be declared locally and based off a global type..

oval:DeprecatedInfoType

< element_mapping >

The element_mapping element is used in documenting which tests, objects, states, and system characteristic items are associated with each other. It provides a way to explicitly and programatically associate the test, object, state, and item definitions.

oval:ElementMapType

< notes >

Element for containing notes; can be replaced using a substitution group.

oval:NotesType

== ElementMapType ==

The ElementMapType is used to document the association between OVAL test, object, state, and item entities.

Child Elements

Table 1: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
test	oval:ElementMapItemType (1..1)	The local name of an OVAL test.
object	oval:ElementMapItemType (0..1)	The local name of an OVAL object.
state	oval:ElementMapItemType (0..1)	The local name of an OVAL state.
item	oval:ElementMapItemType (0..1)	The local name of an OVAL item.

== ElementMapItemType ==

Defines a reference to an OVAL entity using the schema namespace and element name.

Attributes

Table 2: Attributes

Attribute	Type	Desc.
target_namespace	xsd:anyURI (optional)	The target_namespace attributes indicates what XML namespace the element belongs to. If not present, the namespace is that of the document in which the ElementMapItemType instance element appears.

Simple Content: xsd:NCName

== DeprecatedInfoType ==

The DeprecatedInfoType complex type defines a structure that will be used to flag schema-defined constructs as deprecated. It holds information related to the version of OVAL when the construct was deprecated along with a reason and comment.

Child Elements

Table 3: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
version	n/a (1..1)	The required version child element details the version of OVAL in which the construct became deprecated.
reason	xsd:string (1..1)	The required reason child element is used to provide an explanation as to why an item was deprecated and to direct a reader to possible alternative structures within OVAL.
comment	xsd:string (0..1)	The optional comment child element is used to supply additional information regarding the element's deprecated status.

== GeneratorType ==

The GeneratorType complex type defines an element that is used to hold information about when a particular OVAL document was compiled, what version of the schema was used, what tool compiled the document, and what version of that tool was used.

Additional generator information is also allowed although it is not part of the official OVAL Schema. Individual organizations can place generator information that they feel are important and these will be skipped during the validation. All OVAL really cares about is that the stated generator information is there.

Child Elements

Table 4: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
product_name	xsd:string (0..1)	The optional product_name specifies the name of the application used to generate the file. Product names SHOULD be expressed as CPE Names according to the Common Platform Enumeration: Name Matching Specification Version 2.3.
product_version	xsd:string (0..1)	The optional product_version specifies the version of the application used to generate the file.
schema_version	oval:SchemaVersionType (1..unbounded)	The required schema_version specifies the version of the OVAL Schema that the document has been written in and that should be used for validation. The versions for both the Core and any platform extensions used should be declared in separate schema_version elements.
timestamp	xsd:dateTime (1..1)	The required timestamp specifies when the particular OVAL document was compiled. The format for the timestamp is yyyy-mm-ddThh:mm:ss. Note that the timestamp element does not specify when a definition (or set of definitions) was created or modified but rather when the actual XML document that contains the definition was created. For example, the document might have pulled a bunch of existing OVAL Definitions together, each of the definitions having been created at some point in the past. The timestamp in this case would be when the combined document was created.
xsd:any	n/a (0..unbounded)	The Asset Identification specification (http://scap.nist.gov/specifications/ai/) provides a standardized way of reporting asset information across different organizations.Asset Identification elements can hold data useful for identifying what tool, what version of that tool was used, and identify other assets used to compile an OVAL document, such as persons or organizations.To support greater interoperability, an ai:assets element describing assets used to produce an OVAL document may appear at this point in an OVAL document.

== SchemaVersionType ==

The core version MUST match on all platform schema versions.

Attributes

Table 5: Attributes

Attribute	Type	Desc.
platform	xsd:anyURI (optional)	The platform attribute is available to indicate the URI of the target namespace for any platform extension being included. This platform attribute is to be omitted when specifying the core schema version.

Simple Content: oval:SchemaVersionPattern

== MessageType ==

The MessageType complex type defines the structure for which messages are relayed from the data collection engine. Each message is a text string that has an associated level attribute identifying the type of message being sent. These messages could be error messages, warning messages, debug messages, etc. How the messages are used by tools and whether or not they are displayed to the user is up to the specific implementation. Please refer to the description of the MessageLevelEnumeration for more information about each type of message.

Attributes

Table 6: Attributes

Attribute	Type	Desc.
level	oval:MessageLevelEnumeration (optional *default*='info')	(No Description)

Simple Content: xsd:string

== NotesType ==

The NotesType complex type is a container for one or more note child elements. Each note contains some information about the definition or tests that it references. A note may record an unresolved question about the definition or test or present the reason as to why a particular approach was taken.

Child Elements

Table 7: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
note	xsd:string (0..unbounded)	

– CheckEnumeration –

The CheckEnumeration simple type defines acceptable check values, which are used to determine the final result of something based on the results of individual components. When used to define the relationship between objects and states, each check value defines how many of the matching objects (items except those with a status of does not exist) must satisfy the given state for the test to return true. When used to define the relationship between instances of a given entity, the different check values defines how many instances must be true for the entity to return true. When used to define the relationship between entities and multiple variable values, each check value defines how many variable values must be true for the entity to return true.

Table 8: Enumeration Values

Value	Description
all	A value of 'all' means that a final result of true is given if all the individual results under consideration are true.
at least one	A value of 'at least one' means that a final result of true is given if at least one of the individual results under consideration is true.
none exist (Deprecated)	A value of 'none exists' means that a test evaluates to true if no matching object exists that satisfy the data requirements. **Deprecated As Of Version:** 5.3 **Reason:** Replaced by the 'none satisfy' value. In version 5.3 of the OVAL Language, the checking of existence and state were separated into two distinct checks CheckEnumeration (state) and ExistenceEnumeration (existence). Since CheckEnumeration is now used to specify how many objects should satisfy a given state for a test to return true, and no longer used for specifying how many objects must exist for a test to return true, a value of 'none exist' is no longer needed. See the 'none satisfy' value. **Comment:** This value has been deprecated and will be removed in version 6.0 of the language.
none satisfy	A value of 'none satisfy' means that a final result of true is given if none the individual results under consideration are true.
only one	A value of 'only one' means that a final result of true is given if one and only one of the individual results under consideration are true.

Below are some tables that outline how each check attribute effects evaluation. The far left column identifies the check attribute in question. The middle column specifies the different combinations of individual results that the check attribute may bind together. (T=true, F=false, E=error, U=unknown, NE=not evaluated, NA=not applicable) For example, a 1+ under T means that one or more individual results are true, while a 0 under U means that zero individual results are unknown. The last column specifies what the final result would be according to each combination of individual results. Note that if the individual test is negated, then a true result is false and a false result is true, all other results stay as is. "'

|| num of individual results ||

check attr is || || final result is || T | F | E | U | NE | NA ||

——————||————————————||——————

|| 1+ | 0 | 0 | 0 | 0 | 0+ || True || 0+ | 1+ | 0+ | 0+ | 0+ | 0+ || False

ALL || 0+ | 0 | 1+ | 0+ | 0+ | 0+ || Error || 0+ | 0 | 0 | 1+ | 0+ | 0+ || Unknown || 0+ | 0 | 0 | 0 | 1+ | 0+ || Not Evaluated || 0 | 0 | 0 | 0 | 0 | 1+ || Not Applicable

——————||————————————||—————— ‘‘‘

‘‘‘

|| num of individual results ||

check attr is || || final result is || T | F | E | U | NE | NA ||

——————||————————————||——————

|| 1+ | 0+ | 0+ | 0+ | 0+ | 0+ || True || 0 | 1+ | 0 | 0 | 0 | 0+ || False

AT LEAST ONE || 0 | 0+ | 1+ | 0+ | 0+ | 0+ || Error || 0 | 0+ | 0 | 1+ | 0+ | 0+ || Unknown || 0 | 0+ | 0 | 0 | 1+ | 0+ || Not Evaluated || 0 | 0 | 0 | 0 | 0 | 1+ || Not Applicable

——————||————————————||—————— ‘‘‘

‘‘‘

|| num of individual results ||

check attr is || || final result is || T | F | E | U | NE | NA ||

——————||————————————||——————

|| 1 | 0+ | 0 | 0 | 0 | 0+ || True || 2+ | 0+ | 0+ | 0+ | 0+ | 0+ || ** False ** || 0 | 1+ | 0 | 0 | 0 | 0+ || ** False **

ONLY ONE ||0,1 | 0+ | 1+ | 0+ | 0+ | 0+ || Error ||0,1 | 0+ | 0 | 1+ | 0+ | 0+ || Unknown ||0,1 | 0+ | 0 | 0 | 1+ | 0+ || Not Evaluated || 0 | 0 | 0 | 0 | 0 | 1+ || Not Applicable

——————||————————————||—————— ‘‘‘

‘‘‘

|| num of individual results ||

check attr is || || final result is || T | F | E | U | NE | NA ||

——————||————————————||——————

|| 0 | 1+ | 0 | 0 | 0 | 0+ || True || 1+ | 0+ | 0+ | 0+ | 0+ | 0+ || False

NONE SATISFY || 0 | 0+ | 1+ | 0+ | 0+ | 0+ || Error || 0 | 0+ | 0 | 1+ | 0+ | 0+ || Unknown || 0 | 0+ | 0 | 0 | 1+ | 0+ || Not Evaluated || 0 | 0 | 0 | 0 | 0 | 1+ || Not Applicable

——————||————————————||—————— ‘‘‘

– ClassEnumeration –

The ClassEnumeration simple type defines the different classes of definitions. Each class defines a certain intent regarding how an OVAL Definition is written and what that definition is describing. The specified class gives a hint about the definition so a user can know what the definition writer is trying to say. Note that the class does not make a statement about whether a true result is good or bad as this depends on the use of an OVAL Definition. These classes are also used to group definitions by the type of system state they are describing. For example, this allows users to find all the vulnerability (or patch, or inventory, etc) definitions.

Table 9: Enumeration Values

Value	Description
compliance	A compliance definition describes the state of a machine as it complies with a specific policy. A definition of this class will evaluate to true when the system is found to be compliant with the stated policy. Another way of thinking about this is that a compliance definition is stating "the system is compliant if …".
inventory	An inventory definition describes whether a specific piece of software is installed on the system. A definition of this class will evaluate to true when the specified software is found on the system. Another way of thinking about this is that an inventory definition is stating "the software is installed if …".
miscellaneous	The 'miscellaneous' class is used to identify definitions that do not fall into any of the other defined classes.
patch	A patch definition details the machine state of whether a patch executable should be installed. A definition of this class will evaluate to true when the specified patch is missing from the system. Another way of thinking about this is that a patch definition is stating "the patch should be installed if …". Note that word SHOULD is intended to mean more than just CAN the patch executable be installed. In other words, if a more recent patch is already installed then the specified patch might not need to be installed.
vulnerability	A vulnerability definition describes the conditions under which a machine is vulnerable. A definition of this class will evaluate to true when the system is found to be vulnerable with the stated issue. Another way of thinking about this is that a vulnerability definition is stating "the system is vulnerable if …".

– SimpleDatatypeEnumeration –

The SimpleDatatypeEnumeration simple type defines the legal datatypes that are used to describe the values of individual entities that can be represented in a XML string field. The value may have structure and a pattern, but it is represented as string content.

Table 10: Enumeration Values

Value	Description
binary	The binary datatype is used to represent hex-encoded data that is in raw (non-printable) form. This datatype conforms to the W3C Recommendation for binary data meaning that each binary octet is encoded as a character tuple, consisting of two hexadecimal digits {[0-9a-fA-F]} representing the octet code. Expected operations within OVAL for binary values are ‘equals’ and ‘not equal’.
boolean	The boolean datatype represents standard boolean data, either true or false. This datatype conforms to the W3C Recommendation for boolean data meaning that the following literals are legal values: {true, false, 1, 0}. Expected operations within OVAL for boolean values are ‘equals’ and ‘not equal’.
evr_string	The evr_string datatype represents the epoch, version, and release fields as a single version string. It has the form “EPOCH:VERSION-RELEASE”. Comparisons involving this datatype should follow the algorithm of librpm’s rpmvercmp() function. Expected operations within OVAL for evr_string values are ‘equals’, ‘not equal’, ‘greater than’, ‘greater than or equal’, ‘less than’, and ‘less than or equal’.
debian_evr_string	The debian_evr_string datatype represents the epoch, upstream_version, and debian_revision fields, for a Debian package, as a single version string. It has the form “EPOCH:UPSTREAM_VERSION-DEBIAN_REVISION”. Comparisons involving this datatype should follow the algorithm outlined in Chapter 5 of the “Debian Policy Manual” (https://www.debian.org/doc/debian-policy/ch-controlfields.html#s-f-Version). Note that a null epoch is equivalent to a value of ‘0’. An implementation of this is the cmpversions() function in dpkg’s enquiry.c. Expected operations within OVAL for debian_evr_string values are ‘equals’, ‘not equal’, ‘greater than’, ‘greater than or equal’, ‘less than’, and ‘less than or equal’.
fileset_revision	The fileset_revision datatype represents the version string related to filesets in HP-UX. An example would be ‘A.03.61.00’. For more information, see the HP-UX “Software Distributor Administration Guide” (http://h20000.www2.hp.com/bc/docs/support/SupportManual/c01919399/c01919399.pdf). Expected operations within OVAL for fileset_version values are ‘equals’, ‘not equal’, ‘greater than’, ‘greater than or equal’, ‘less than’, and ‘less than or equal’.

– ComplexDatatypeEnumeration –

The ComplexDatatypeEnumeration simple type defines the complex legal datatypes that are supported in OVAL. These datatype describe the values of individual entities where the entity has some complex structure beyond simple string like content.

Table 11: Enumeration Values

Value	Description
record	The record datatype describes an entity with structured set of named fields and values as its content. The only allowed operation within OVAL for record values is 'equals'. Note that the record datatype is not currently allowed when using variables.

– DatatypeEnumeration –

The DatatypeEnumeration simple type defines the legal datatypes that are used to describe the values of individual entities. A value should be interpreted according to the specified type. This is most important during comparisons. For example, is '21' less than '123'? will evaluate to true if the datatypes are 'int', but will evaluate to 'false' if the datatypes are 'string'. Another example is applying the 'equal' operation to '1.0.0.0' and '1.0'. With datatype 'string' they are not equal, with datatype 'version' they are.

** Union of **oval:SimpleDatatypeEnumeration, oval:ComplexDatatypeEnumeration .. _ExistenceEnumeration:

– ExistenceEnumeration –

The ExistenceEnumeration simple type defines acceptable existence values, which are used to determine a result based on the existence of individual components. The main use for this is for a test regarding the existence of objects on the system. Its secondary use is for a state regarding the existence of entities in corresponding items.

Table 12: Enumeration Values

Value	Description
all_exist	When used in the context of an OVAL state entity's check_existence attribute, a value of 'all_exist' means that every item entity for an object defined by the description exists on the system. When used in the context of an OVAL test's check_existence attribute, this value is equivalent to 'at_least_one_exists' because non-existent items have no impact upon evaluation.
any_exist	A value of 'any_exist' means that zero or more objects defined by the description exist on the system.
at_least_one_exists	A value of 'at_least_one_exists' means that at least one object defined by the description exists on the system.
none_exist	A value of 'none_exist' means that none of the objects defined by the description exist on the system.
only_one_exists	A value of 'only_one_exists' means that only one object defined by the description exists on the system.

Below are some tables that outline how each ExistenceEnumeration value effects evaluation of a given test. Note that this is related to the existence of an object(s) and not the object(s) compliance with a state. The left column identifies the ExistenceEnumeration value in question. The middle column specifies the different combinations of individual item status values that have been found in the system characteristics file related to the given object. (EX=exists, DE=does not exist, ER=error, NC=not collected) For example, a 1+ under EX means that one or more individual item status attributes are set to exists, while a 0 under NC means that zero individual item status attributes are set to not collected. The last column specifies what the result of the existence piece would be according to each combination of individual item status values. ‘‘‘

|| item status value count ||

attr value || || existence piece is || EX | DE | ER | NC ||

——————||——————||——————

|| 1+ | 0 | 0 | 0 || True || 0 | 0 | 0 | 0 || False || 0+ | 1+ | 0+ | 0+ || False

all_exist || 0+ | 0 | 1+ | 0+ || Error || 0+ | 0 | 0 | 1+ || Unknown || – | – | – | – || Not Evaluated || – | – | – | – || Not Applicable

——————||——————||—————— ‘‘‘

‘‘‘

|| item status value count ||

attr value || || existence piece is || EX | DE | ER | NC ||

——————————||——————————————||—————————

|| 0+ | 0+ | 0 | 0+ || True || 1+ | 0+ | 1+ | 0+ || True || – | – | – | – || False

any_exist || 0 | 0+ | 1+ | 0+ || Error || – | – | – | – || Unknown || – | – | – | – || Not Evaluated || – | – | – | – || Not Applicable

——————————||——————————————||————————— ‘‘‘

‘‘‘

|| item status value count ||

attr value || || existence piece is || EX | DE | ER | NC ||

——————————||——————————————||————————— || 1+ | 0+ | 0+ | 0+ || True || 0 | 0+ | 0 | 0 || False

at_least_one_exists || 0 | 0+ | 1+ | 0+ || Error || 0 | 0+ | 0 | 1+ || Unknown || – | – | – | – || Not Evaluated || – | – | – | – || Not Applicable

——————————||——————————————||————————— ‘‘‘

‘‘‘

|| item status value count ||

attr value || || existence piece is || EX | DE | ER | NC ||

——————————||——————————————||—————————

|| 0 | 0+ | 0 | 0 || True || 1+ | 0+ | 0+ | 0+ || False

none_exist || 0 | 0+ | 1+ | 0+ || Error || 0 | 0+ | 0 | 1+ || Unknown || – | – | – | – || Not Evaluated || – | – | – | – || Not Applicable

——————————||——————————————||————————— ‘‘‘

‘‘‘

|| item status value count ||

attr value || || existence piece is || EX | DE | ER | NC ||

——————————||——————————————||—————————

|| 1 | 0+ | 0 | 0 || True || 2+ | 0+ | 0+ | 0+ || False || 0 | 0+ | 0 | 0 || False

only_one_exists || 0,1 | 0+ | 1+ | 0+ || Error || 0,1 | 0+ | 0 | 1+ || Unknown || – | – | – | – || Not Evaluated || – | – | – | – || Not Applicable

——————————||——————————————||————————— ‘‘‘

– FamilyEnumeration –

The FamilyEnumeration simple type is a listing of families that OVAL supports at this time. Since new family values can only be added with new version of the schema, the value of ‘undefined’ is to be used when the desired family is not available. Note that use of the undefined family value does not target all families, rather it means that some family other than one of the defined values is targeted.

Table 13: Enumeration Values

Value	Description
android	The android value describes the Android mobile operating system.
asa	The asa value describes the Cisco ASA security devices.
apple_ios	The apple_ios value describes the iOS mobile operating system.
catos	The catos value describes the Cisco CatOS operating system.
ios	The ios value describes the Cisco IOS operating system.
iosxe	The iosxe value describes the Cisco IOS XE operating system.
junos	The junos value describes the Juniper JunOS operating system.
macos	The macos value describes the Mac operating system.
pixos	The pixos value describes the Cisco PIX operating system.
undefined	The undefined value is to be used when the desired family is not available.
unix	The unix value describes the UNIX operating system.
vmware_infrastructure	The vmware_infrastructure value describes VMWare Infrastructure.
windows	The windows value describes the Microsoft Windows operating system.

– MessageLevelEnumeration –

The MessageLevelEnumeration simple type defines the different levels associated with a message. There is no specific criteria about which messages get assigned which level. This is completely arbitrary and up to the content producer to decide what is an error message and what is a debug message.

Table 14: Enumeration Values

Value	Description
debug	Debug messages should only be displayed by a tool when run in some sort of verbose mode.
error	Error messages should be recorded when there was an error that did not allow the collection of specific data.
fatal	A fatal message should be recorded when an error causes the failure of more than just a single piece of data.
info	Info messages are used to pass useful information about the data collection to a user.
warning	A warning message reports something that might not correct but information was still collected.

– OperationEnumeration –

The OperationEnumeration simple type defines acceptable operations. Each operation defines how to compare entities against their actual values.

Table 15: Enumeration Values

Value	Description
equals	The ‘equals’ operation returns true if the actual value on the system is equal to the stated entity. When the specified datatype is a string, this results in a case-sensitive comparison.
not equal	The ‘not equal’ operation returns true if the actual value on the system is not equal to the stated entity. When the specified datatype is a string, this results in a case-sensitive comparison.
case insensitive equals	The ‘case insensitive equals’ operation is meant for string data and returns true if the actual value on the system is equal (using a case insensitive comparison) to the stated entity.
case insensitive not equal	The ‘case insensitive not equal’ operation is meant for string data and returns true if the actual value on the system is not equal (using a case insensitive comparison) to the stated entity.
greater than	The ‘greater than’ operation returns true if the actual value on the system is greater than the stated entity.
less than	The ‘less than’ operation returns true if the actual value on the system is less than the stated entity.
greater than or equal	The ‘greater than or equal’ operation returns true if the actual value on the system is greater than or equal to the stated entity.
less than or equal	The ‘less than or equal’ operation returns true if the actual value on the system is less than or equal to the stated entity.
bitwise and	The ‘bitwise and’ operation is used to determine if a specific bit is set. It returns true if performing a BITWISE AND with the binary representation of the stated entity against the binary representation of the actual value on the system results in a binary value that is equal to the binary representation of the stated entity. For example, assuming a datatype of ‘int’, if the actual integer value of the setting on your machine is 6 (same as 0110 in binary), then performing a ‘bitwise and’

– OperatorEnumeration –

The OperatorEnumeration simple type defines acceptable operators. Each operator defines how to evaluate multiple arguments.

Table 16: Enumeration Values

Value	Description
AND	The AND operator produces a true result if every argument is true. If one or more arguments are false, the result of the AND is false. If one or more of the arguments are unknown, and if none of the arguments are false, then the AND operator produces a result of unknown.
ONE	The ONE operator produces a true result if one and only one argument is true. If there are more than argument is true (or if there are no true arguments), the result of the ONE is false. If one or more of the arguments are unknown, then the ONE operator produces a result of unknown.
OR	The OR operator produces a true result if one or more arguments is true. If every argument is false, the result of the OR is false. If one or more of the arguments are unknown and if none of arguments are true, then the OR operator produces a result of unknown.
XOR	XOR is defined to be true if an odd number of its arguments are true, and false otherwise. If any of the arguments are unknown, then the XOR operator produces a result of unknown.

Below are some tables that outline how each operator effects evaluation. The far left column identifies the operator in question. The middle column specifies the different combinations of individual results that the operator may bind together. (T=true, F=false, E=error, U=unknown, NE=not evaluated, NA=not applicable) For example, a 1+ under T means that one or more individual results are true, while a 0 under U means that zero individual results are unknown. The last column specifies what the final result would be according to each combination of individual results. Note that if the individual test is negated, then a true result is false and a false result is true, all other results stay as is. ‘‘‘

|| num of individual results ||

operator is || || final result is || T | F | E | U | NE | NA ||

——————||——————————————||——————————

|| 1+ | 0 | 0 | 0 | 0 | 0+ || True || 0+ | 1+ | 0+ | 0+ | 0+ | 0+ || False

AND || 0+ | 0 | 1+ | 0+ | 0+ | 0+ || Error || 0+ | 0 | 0 | 1+ | 0+ | 0+ || Unknown || 0+ | 0 | 0 | 0 | 1+ | 0+ || Not Evaluated || 0 | 0 | 0 | 0 | 0 | 1+ || Not Applicable

——————||——————————||—————— ‘‘‘

‘‘‘

|| num of individual results ||

operator is || || final result is || T | F | E | U | NE | NA ||

——————||——————————||——————

|| 1 | 0+ | 0 | 0 | 0 | 0+ || True || 2+ | 0+ | 0+ | 0+ | 0+ | 0+ || ** False ** || 0 | 1+ | 0 | 0 | 0 | 0+ || ** False **

ONE ||0,1 | 0+ | 1+ | 0+ | 0+ | 0+ || Error ||0,1 | 0+ | 0 | 1+ | 0+ | 0+ || Unknown ||0,1 | 0+ | 0 | 0 | 1+ | 0+ || Not Evaluated || 0 | 0 | 0 | 0 | 0 | 1+ || Not Applicable

——————||——————————||—————— ‘‘‘

‘‘‘

|| num of individual results ||

operator is || || final result is || T | F | E | U | NE | NA ||

——————||——————————||——————

|| 1+ | 0+ | 0+ | 0+ | 0+ | 0+ || True || 0 | 1+ | 0 | 0 | 0 | 0+ || False

OR || 0 | 0+ | 1+ | 0+ | 0+ | 0+ || Error || 0 | 0+ | 0 | 1+ | 0+ | 0+ || Unknown || 0 | 0+ | 0 | 0 | 1+ | 0+ || Not Evaluated || 0 | 0 | 0 | 0 | 0 | 1+ || Not Applicable

——————||——————————||—————— ‘‘‘

‘‘‘

|| num of individual results ||

operator is || || final result is || T | F | E | U | NE | NA ||

——————||——————————||——————

||odd | 0+ | 0 | 0 | 0 | 0+ || True ||even| 0+ | 0 | 0 | 0 | 0+ || False

XOR || 0+ | 0+ | 1+ | 0+ | 0+ | 0+ || Error || 0+ | 0+ | 0 | 1+ | 0+ | 0+ || Unknown || 0+ | 0+ | 0 | 0 | 1+ | 0+ || Not Evaluated || 0 | 0 | 0 | 0 | 0 | 1+ || Not Applicable

——————||——————————||—————— ‘‘‘

– DefinitionIDPattern –

Define the format for acceptable OVAL Definition ids. An urn format is used with the id starting with the word oval followed by a unique string, followed by the three letter code ‘def’, and ending with an integer.

oval:[**A-Za-z0-9_**-.]+:def:[1-9][0-9]*.. _ObjectIDPattern:

– ObjectIDPattern –

Define the format for acceptable OVAL Object ids. An urn format is used with the id starting with the word oval followed by a unique string, followed by the three letter code ‘obj’, and ending with an integer.

oval:[**A-Za-z0-9_**-.]+:obj:[1-9][0-9]*.. _StateIDPattern:

– StateIDPattern –

Define the format for acceptable OVAL State ids. An urn format is used with the id starting with the word oval followed by a unique string, followed by the three letter code ‘ste’, and ending with an integer.

oval:[**A-Za-z0-9_**-.]+:ste:[1-9][0-9]*.. _TestIDPattern:

– TestIDPattern –

Define the format for acceptable OVAL Test ids. An urn format is used with the id starting with the word oval followed by a unique string, followed by the three letter code ‘tst’, and ending with an integer.

oval:[**A-Za-z0-9_**-.]+:tst:[1-9][0-9]*.. _VariableIDPattern:

– VariableIDPattern –

Define the format for acceptable OVAL Variable ids. An urn format is used with the id starting with the word oval followed by a unique string, followed by the three letter code ‘var’, and ending with an integer.

oval:[**A-Za-z0-9_**-.]+:var:[1-9][0-9]*.. _ItemIDPattern:

– ItemIDPattern –

Define the format for acceptable OVAL Item ids. The format is an integer. An item id is used to identify the different items found in an OVAL System Characteristics file.

– SchemaVersionPattern –

Define the format for acceptable OVAL Language version strings.

[0-9]+.[0-9]+(.[0-9]+)?(:[0-9]+.[0-9]+(.[0-9]+)?)?______________

– EmptyStringType –

The EmptyStringType simple type is a restriction of the built-in string simpleType. The only allowed string is the empty string with a length of zero. This type is used by certain elements to allow empty content when non-string data is accepted. See the EntityIntType in the OVAL Definition Schema for an example of its use.

– NonEmptyStringType –

The NonEmptyStringType simple type is a restriction of the built-in string simpleType. Empty strings are not allowed. This type is used by comment attributes where an empty value is not allowed.

Open Vulnerability and Assessment Language: Core Definition

- Schema: Core Definition
- Version: 5.11.2
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the core schema for encoding Open Vulnerability and Assessment Language (OVAL) Definitions. Some of the objects defined here are extended and enhanced by individual component schemas, which are described in separate documents. Each of the elements, types, and attributes that make up the Core Definition Schema are described in detail and should provide the information necessary to understand what each 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 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.cisecurity.org.

< oval_definitions >

The oval_definitions element is the root of an OVAL Definition Document. Its purpose is to bind together the major sections of a document - generator, definitions, tests, objects, states, and variables - which are the children of the root element.

Child Elements

Table 17: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
generator	oval:GeneratorType (1..1)	The required generator section provides information about when the definition file was compiled and under what version.
definitions	oval-def:DefinitionsType (0..1)	The optional definitions section contains 1 or more definitions.
tests	oval-def:TestsType (0..1)	The optional tests section contains 1 or more tests.
objects	oval-def:ObjectsType (0..1)	The optional objects section contains 1 or more objects.
states	oval-def:StatesType (0..1)	The optional states section contains 1 or more states.
variables	oval-def:VariablesType (0..1)	The optional variables section contains 1 or more variables.
ds:Signature	n/a (0..1)	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. More information about the official W3C Recommendation regarding XML digital signatures can be found at http://www.w3.org/TR/xmldsig-core/.

< notes > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.11.1
- Reason: Replaced by the oval:notes element.
- Comment: This object has been deprecated and may be removed in a future version of the language.

The notes element is a container for one or more note child elements. It exists for backwards-compatibility purposes, for the pre-5.11.0 oval-def:NotesType, which has been replaced by the oval:notes element in 5.11.1.

Extends: oval:NotesType

Child Elements

Table 18: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
note	xsd:string (0..unbounded)	

== DefinitionsType ==

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

Child Elements

Table 19: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
oval-def:definition	n/a (1..unbounded)	

< definition >

The definition element represents the globally defined element of type DefinitionType. For more information please see the documentation on the DefinitionType.

oval-def:DefinitionType

== DefinitionType ==

The DefinitionType defines a single OVAL Definition. A definition is the key structure in OVAL. It is analogous to the logical sentence or proposition: if a computer's state matches the configuration parameters laid out in the criteria, then that computer exhibits the state described. The DefinitionType contains a section for various metadata related elements that describe the definition. This includes a description, version, affected system types, and reference information.

The notes section of a definition should be used to hold information that might be helpful to someone examining the technical aspects of the definition. For example, why certain tests have been included in the criteria, or maybe a link to where further information can be found. The DefinitionType also (unless the definition is deprecated) contains a criteria child element that joins individual tests together with a logical operator to specify the specific computer state being described.

The required id attribute is the OVAL-ID of the Definition. The form of an OVAL-ID must follow the specific format described by the oval:DefinitionIDPattern. The required version attribute holds the current version of the definition. Versions are integers, starting at 1 and incrementing every time a definition is modified. The required class attribute indicates the specific class to which the definition belongs. The class gives a hint to a user so they can know what the definition writer is trying to say. See the definition of oval-def:ClassEnumeration for more information about the different valid classes. The optional deprecated attribute signifies that an id is no longer to be used or referenced but the information has been kept around for historic purposes.

When the deprecated attribute is set to true, the definition is considered to be deprecated. The criteria child element of a deprecated definition is optional. If a deprecated definition does not contain a criteria child element, the definition must evaluate to "not evaluated". If a deprecated definition contains a criteria child element, an interpreter should evaluate the definition as if it were not deprecated, but an interpreter may evaluate the definition to "not evaluated".

Attributes

Table 20: Attributes

Attribute	Type	Desc.
id	oval:DefinitionIDPattern (required)	(No Description)
version	xsd:nonNegativeInteger (required)	(No Description)
class	oval:ClassEnumeration (required)	(No Description)
deprecated	xsd:boolean (optional *default*='false')	(No Description)

Child Elements

Table 21: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
ds:Signature	n/a (0..1)	
metadata	oval-def:MetadataType (1..1)	
oval:notes	n/a (0..1)	
criteria	oval-def:CriteriaType (0..1)	

== MetadataType ==

The MetadataType complex type contains all the metadata available to an OVAL Definition. This metadata is for informational purposes only and is not part of the criteria used to evaluate machine state. The required title child element holds a short string that is used to quickly identify the definition to a human user. The affected metadata item contains information about the system(s) for which the definition has been written. Remember that this is just metadata and not part of the criteria. Please refer to the AffectedType description for more information. The required description element contains a textual description of the configuration state being addressed by the OVAL Definition. In the case of a definition from the vulnerability class, the reference is usually the Common Vulnerability and Exposures (CVE) Identifier, and this description field corresponds with the CVE description.

Additional metadata is also allowed although it is not part of the official OVAL Schema. Individual organizations can place metadata items that they feel are important and these will be skipped during the validation. All OVAL really cares about is that the stated metadata items are there.

Child Elements

Table 22: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
title	xsd:string (1..1)	
affected	oval-def:AffectedType (0..unbounded)	
reference	oval-def:ReferenceType (0..unbounded)	
description	xsd:string (1..1)	
xsd:any	n/a (0..unbounded)	

== AffectedType ==

Each OVAL Definition is written to evaluate a certain type of system(s). The family, platform(s), and product(s) of this target are described by the AffectedType whose main purpose is to provide hints for tools using OVAL Definitions. For instance, to help a reporting tool only use Windows definitions, or to preselect only Red Hat definitions to be evaluated. Note, the inclusion of a particular platform or product does not mean the definition is physically checking for the existence of the platform or product. For the actual test to be performed, the correct test must still be included in the definition's criteria section.

The AffectedType complex type details the specific system, application, subsystem, library, etc. for which a definition has been written. If a definition is not tied to a specific product, then this element should not be included. The absence of the platform or product element can be thought of as definition applying to all platforms or products. The inclusion of a particular platform or product does not mean the definition is physically checking for the existence of the platform or product. For the actual test to be performed, the correct test must still be included in the definition's criteria section. To increase the utility of this element, care should be taken when assigning and using strings for product names. The schema places no restrictions on the values that can be assigned, potentially leading to many different representations of the same value. For example, 'Internet Explorer' and 'IE' might be used to refer to the same product. The current convention is to fully spell out all terms, and avoid the use of abbreviations at all costs.

Please note that the AffectedType will change in future versions of OVAL in order to support the Common Platform Enumeration (CPE).

Attributes

Table 23: Attributes

Attribute	Type	Desc.
family	oval:FamilyEnumeration (required)	(No Description)

Child Elements

Table 24: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
platform	xsd:string (0..unbounded)	
product	xsd:string (0..unbounded)	

== ReferenceType ==

The ReferenceType complex type links the OVAL Definition to a definitive external reference. For example, CVE Identifiers are used for referencing vulnerabilities. The intended purpose for this reference is to link the definition to a variety of other sources that address the same issue being specified by the OVAL Definition.

The required source attribute specifies where the reference is coming from. In other words, it identifies the reference repository being used. The required ref_id attribute is the external id of the reference. The optional ref_url attribute is the URL to the reference.

Attributes

Table 25: Attributes

Attribute	Type	Desc.
source	xsd:string (required)	(No Description)
ref_id	xsd:string (required)	(No Description)
ref_url	xsd:anyURI (optional)	(No Description)

== CriteriaType ==

The CriteriaType complex type describes a container for a set of sub criteria, criteria, criterion, or extend_definition elements 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 and 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 the criteria as a whole should be negated during analysis. For example, consider a criteria that evaluates to TRUE if certain software is installed. By negating this test, it now evaluates to TRUE if the software is NOT installed. The optional comment attribute provides a short description of the criteria.

The optional applicability_check attribute provides a Boolean flag that when true indicates that the criteria is being used to determine whether the OVAL Definition applies to a given system.

Attributes

Table 26: Attributes

Attribute	Type	Desc.
applicability_check	xsd:boolean (optional)	(No Description)
operator	oval:OperatorEnumeration (optional *default*='AND')	(No Description)
negate	xsd:boolean (optional *default*='false')	(No Description)
comment	oval:NonEmptyStringType (optional)	(No Description)

Child Elements

Table 27: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
criteria	oval-def:CriteriaType (1..unbounded)	
criterion	oval-def:CriterionType (1..unbounded)	
extend_definition	oval-def:ExtendDefinitionType (1..unbounded)	

== CriterionType ==

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

The required test_ref attribute is the actual id of the test being referenced. 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 optional comment attribute provides a short description of the specified test and should mirror the comment attribute of the actual test.

The optional applicability_check attribute provides a Boolean flag that when true indicates that the criterion is being used to determine whether the OVAL Definition applies to a given system.

Attributes

Table 28: Attributes

Attribute	Type	Desc.
applicability_check	xsd:boolean (optional)	(No Description)
test_ref	oval:TestIDPattern (required)	(No Description)
negate	xsd:boolean (optional *default*='false')	(No Description)
comment	oval:NonEmptyStringType (optional)	(No Description)

== ExtendDefinitionType ==

The ExtendDefinitionType complex type allows existing definitions to be extended by another definition. This works by evaluating the extended definition and then using the result within the logical context of the extending definition.

The required definition_ref attribute is the actual id of the definition being extended. 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 certainsoftware is installed. By negating the definition, it now evaluates to TRUE if the software is NOT installed. The optional comment attribute provides a short description of the specified definition and should mirror the title metadata of the extended definition.

The optional applicability_check attribute provides a Boolean flag that when true indicates that the extend_definition is being used to determine whether the OVAL Definition applies to a given system.

Attributes

Table 29: Attributes

Attribute	Type	Desc.
applicability_check	xsd:boolean (optional)	(No Description)
definition_ref	oval:DefinitionIDPattern (required)	(No Description)
negate	xsd:boolean (optional *default*='false')	(No Description)
comment	oval:NonEmptyStringType (optional)	(No Description)

== TestsType ==

The TestsType complex type is a container for one or more test child elements. Each test element describes a single OVAL Test. Please refer to the description of the TestType for more information about an individual test.

Child Elements

Table 30: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
oval-def:test	n/a (1..unbounded)	

< test >

The test element is an abstract element that is meant to be extended (via substitution groups) by the individual tests found in the component schemas. An OVAL Test is used to compare an object(s) against a defined state. An actual test element is not valid. The use of this abstract class simplifies the OVAL schema by allowing individual tests to inherit the optional notes child element, and the id and comment attributes from the base TestType. Please refer to the description of the TestType complex type for more information.

oval-def:TestType

== TestType ==

The base type of every test includes an optional notes element and several attributes. The notes section of a test should be used to hold information that might be helpful to someone examining the technical aspects of the test. For example, why certain values have been used by the test, or maybe a link to where further information can be found. Please refer to the description of the NotesType complex type for more information about the notes element. The required comment attribute provides a short description of the test. The optional deprecated attribute signifies that an id is no longer to be used or referenced but the information has been kept around for historic purposes.

The required id attribute uniquely identifies each test, and must conform to the format specified by the TestIdPattern simple type. The required version attribute holds the current version of the test. Versions are integers, starting at 1 and incrementing every time a test is modified.

The optional check_existence attribute specifies how many items in the set defined by the OVAL Object must exist for the test to evaluate to true. The default value for this attribute is 'at_least_one_exists' indicating that by default the test may evaluate to true if at least one item defined by the OVAL Object exists on the system. For example, if a value of 'all_exist' is given, every item defined by the OVAL Object must exist on the system for the test to evaluate to true. If the OVAL Object uses a variable reference, then every value of that variable must exist. Note that a pattern match defines a unique set of matching items found on a system. So when check_existence = 'all_exist' and a regex matches anything on a system the test will evaluate to true (since all matching objects on the system were found on the system). When check_existence = 'all_exist' and a regex does not match anything on a system the test will evaluate to false.

The required check attribute specifies how many items in the set defined by the OVAL Object (ignoring items with a status of Does Not Exist) must satisfy the state requirements. For example, should the test check that all matching files have a specified version or that at least one file has the specified version? The valid check values are explained in the description of the CheckEnumeration simple type. Note that if the test does not contain any references to OVAL States, then the check attribute has no meaning and can be ignored during evaluation.

An OVAL Test evaluates to true if both the check_existence and check attributes are satisfied during evaluation. The evaluation result for a test is determined by first evaluating the check_existence attribute. If the result of evaluating the check_existence attribute is true then the check attribute is evaluated. An interpreter may choose to always evaluate both the check_existence and the check attributes, but once the check_existence attribute evaluation has resulted in false the overall test result after evaluating the check attribute will not be affected.

The optional state_operator attribute provides the logical operator that combines the evaluation results from each referenced state on a per item basis. Each matching item is compared to each referenced state. The result of comparing each state to a single item is combined based on the specified state_operator value to determine one result for each item. Finally, the results for each item are combined based on the specified check value. Note that if the test does not contain any references to OVAL States, then the state_operator attribute has no meaning and can be ignored during evaluation. Referencing multiple states in one test allows ranges of possible values to be expressed. For example, one state can check that a value greater than 8 is found and another state can check that a value of less than 16 is found. In this example the referenced states are combined with a state_operator = 'AND' indicating that the conditions of all referenced states must be satisfied and that the value must be between 8 AND 16. The valid state_operation values are explained in the description of the OperatorEnumeration simple type.

Attributes

Table 31: Attributes

Attribute	Type	Desc.
id	oval:TestIDPattern (required)	(No Description)
version	xsd:nonNegativeInteger (required)	(No Description)
check_existence	oval:ExistenceEnumeration (optional *default*='at_least_one_exists')	(No Description)
check	oval:CheckEnumeration (required)	(No Description)
state_operator	oval:OperatorEnumeration (optional *default*='AND')	(No Description)
comment	oval:NonEmptyStringType (required)	(No Description)
deprecated	xsd:boolean (optional *default*='false')	(No Description)

Child Elements

Table 32: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
ds:Signature	n/a (0..1)	
oval:notes	n/a (0..1)	

== ObjectRefType ==

The ObjectRefType complex type defines an object reference to be used by OVAL Tests that are defined in the component schemas. The required object_ref attribute specifies the id of the OVAL Object being referenced.

Attributes

Table 33: Attributes

Attribute	Type	Desc.
object_ref	oval:ObjectIDPattern (required)	(No Description)

== StateRefType ==

The StateRefType complex type defines a state reference to be used by OVAL Tests that are defined in the component schemas. The required state_ref attribute specifies the id of the OVAL State being referenced.

Attributes

Table 34: Attributes

Attribute	Type	Desc.
state_ref	oval:StateIDPattern (required)	(No Description)

== ObjectsType ==

The ObjectsType complex type is a container for one or more object child elements. Each object element provides details that define a unique set of matching items to be used by an OVAL Test. Please refer to the description of the object element for more information about an individual object.

Child Elements

Table 35: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
oval-def:object	n/a (1..unbounded)	

< object >

The object element is an abstract element that is meant to be extended (via substitution groups) by the objects found in the component schemas. An actual object element is not valid. The use of this abstract element simplifies the OVAL schema by allowing individual objects to inherit any common elements and attributes from the base ObjectType. Please refer to the description of the ObjectType complex type for more information.

An object is used to identify a set of items to collect. The author of a schema object must define sufficient object entities to allow a user to identify a unique item to be collected.

A simple object typically results in a single file, process, etc being identified. But through the use of pattern matches, sets, and variables, multiple matching items can be identified. The set of items matching the object can then be used by an OVAL test and compared against an OVAL state.

oval-def:ObjectType

== ObjectType ==

The base type of every object includes an optional notes element. The notes element of an object should be used to hold information that might be helpful to someone examining the technical aspects of the object. For example, why certain values have been used, or maybe a link to where further information can be found. Please refer to the description of the NotesType complex type for more information about the notes element.

The required id attribute uniquely identifies each object, and must conform to the format specified by the ObjectIdPattern simple type. The required version attribute holds the current version of the object element. Versions are integers, starting at 1 and incrementing every time an object is modified. The optional comment attribute provides a short description of the object. The optional deprecated attribute signifies that an id is no longer to be used or referenced but the information has been kept around for historic purposes.

Attributes

Table 36: Attributes

Attribute	Type	Desc.
id	oval:ObjectIDPattern (required)	(No Description)
version	xsd:nonNegativeInteger (required)	(No Description)
comment	oval:NonEmptyStringType (optional)	(No Description)
deprecated	xsd:boolean (optional *default*='false')	(No Description)

Child Elements

Table 37: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
ds:Signature	n/a (0..1)	
oval:notes	n/a (0..1)	

< set >

The set element enables complex objects to be described. It is a recursive element in that each set element can contain additional set elements as children. Each set element defines characteristics that produce a matching unique set of items. This set of items is defined by one or two references to OVAL Objects that provide the criteria needed to collect a set of system items. These items can have one or more filters applied to allow a subset of those items to be specifically included or excluded from the overall set of items.

The set element's object_reference refers to an existing OVAL Object. The set element's filter element provides a reference to an existing OVAL State and includes an optional action attribute. The filter's action attribute allows the author to specify whether matching items should be included or excluded from the overall set. The default filter action is to exclude all matching items. In other words, the filter can be thought of filtering items out by default.

Each filter is applied to the items identified by each OVAL Object before the set_operator is applied. For example, if an object_reference points to an OVAL Object that identifies every file in a certain directory, a filter might be set up to limit the object set to only those files with a size less than 10 KB. If multiple filters are provided, then each filter is applied to the set of items identified by the OVAL Object. Care must be taken to ensure that conflicting filters are not applied. It is possible to exclude all items with a size of 10 KB and then include only items with a size of 10 KB. This example would result in the empty set.

The required set_operator attribute defines how different child sets are combined to form the overall unique set of objects. For example, does one take the union of different sets or the intersection? For a description of the valid values please refer to the SetOperatorEnumeration simple type.

Child Elements

Table 38: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object_reference	oval:ObjectIDPattern (1..2)	
oval-def:filter	n/a (0..unbounded)	

< filter >

The filter element provides a reference to an existing OVAL State and includes an optional action attribute. The action attribute is used to specify whether items that match the referenced OVAL State will be included in the resulting set or excluded from the resulting set.

== StatesType ==

The StatesType complex type is a container for one or more state child elements. Each state provides details about specific characteristics that can be used during an evaluation of an object. Please refer to the description of the state element for more information about an individual state.

Child Elements

Table 39: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
oval-def:state	n/a (1..unbounded)	

< state >

The state element is an abstract element that is meant to be extended (via substitution groups) by the states found in the component schemas. An actual state element is not valid. The use of this abstract class simplifies the OVAL schema by allowing individual states to inherit the optional notes child element, and the id and operator attributes from the base StateType. Please refer to the description of the StateType complex type for more information.

An OVAL State is a collection of one or more characteristics pertaining to a specific object type. The OVAL State is used by an OVAL Test to determine if a unique set of items identified on a system meet certain characteristics.

oval-def:StateType

== StateType ==

The base type of every state includes an optional notes element and two attributes. The notes section of a state should be used to hold information that might be helpful to someone examining the technical aspects of the state. For example, why certain values have been used by the state, or maybe a link to where further information can be found. Please refer to the description of the NotesType complex type for more information about the notes element.

The required id attribute uniquely identifies each state, and must conform to the format specified by the StateIdPattern simple type. The required version attribute holds the current version of the state. Versions are integers, starting at 1 and incrementing every time a state is modified. The required operator attribute provides the logical operator that binds the different characteristics inside a state together. The optional comment attribute provides a short description of the state. The optional deprecated attribute signifies that an id is no longer to be used or referenced but the information has been kept around for historic purposes.

When evaluating a particular state against an object, one should evaluate each individual entity separately. The individual results are then combined by the operator to produce an overall result. This process holds true even when there are multiple instances of the same entity. Evaluate each instance separately, taking the entity check attribute into account, and then combine everything using the operator.

Attributes

Table 40: Attributes

Attribute	Type	Desc.
id	oval:StateIDPattern (required)	(No Description)
version	xsd:nonNegativeInteger (required)	(No Description)
operator	oval:OperatorEnumeration (optional *default*='AND')	(No Description)
comment	oval:NonEmptyStringType (optional)	(No Description)
deprecated	xsd:boolean (optional *default*='false')	(No Description)

Child Elements

Table 41: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
ds:Signature	n/a (0..1)	
oval:notes	n/a (0..1)	

== VariablesType ==

The VariablesType complex type is a container for one or more variable child elements. Each variable element is a way to define one or more values to be obtained at the time a definition is evaluated.

Child Elements

Table 42: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
oval-def:variable	n/a (1..unbounded)	

< variable >

The variable element is an abstract element that is meant to be extended (via substitution groups) by the different types of variables. An actual variable element is not valid. The different variable types describe different sources for obtaining a value(s) for the variable. There are currently three types of variables; local, external, and constant. Please refer to the description of each one for more specific information. The value(s) of a variable is treated as if it were inserted where referenced. One of the main benefits of variables is that they allow tests to evaluate user-defined policy. For example, an OVAL Test might check to see if a password is at least a certain number of characters long, but this number depends upon the individual policy of the user. To solve this, the test for password length can be written to refer to a variable element that defines the length.

If a variable defines a collection of values, any entity that references the variable will evaluate to true depending on the value of the var_check attribute. For example, if an entity 'size' with an operation of 'less than' references a variable that returns five different integers, and the var_check attribute has a value of 'all', then the 'size' entity returns true only if the actual size is less than each of the five integers defined by the variable. If a variable does not return any value, then an error should be reported during OVAL analysis.

oval-def:VariableType

== VariableType ==

The VariableType complex type defines attributes associated with each OVAL Variable. The required id attribute uniquely identifies each variable, and must conform to the format specified by the VariableIDPattern simple type. The required version attribute holds the current version of the variable. Versions are integers, starting at 1 and incrementing every time a variable is modified. The required comment attribute provides a short description of the variable. The optional deprecated attribute signifies that an id is no longer to be used or referenced but the information has been kept around for historic purposes.

The required datatype attribute specifies the type of value being defined. The set of values identified by a variable must comply with the specified datatype, otherwise an error should be reported. Please see the DatatypeEnumeration for details about each valid datatype. For example, if the datatype of the variable is specified as boolean then the value(s) returned by the component / function should be "true", "false", "1", or "0".

Note that the 'record' datatype is not permitted on variables. The notes section of a variable should be used to hold information that might be helpful to someone examining the technical aspects of the variable. Please refer to the description of the NotesType complex type for more information about the notes element.

Attributes

Table 43: Attributes

Attribute	Type	Desc.
id	oval:VariableIDPattern (required)	(No Description)
version	xsd:nonNegativeInteger (required)	(No Description)
datatype	oval:SimpleDatatypeEnumeration (required)	Note that the 'record' datatype is not permitted on variables.
comment	oval:NonEmptyStringType (required)	(No Description)
deprecated	xsd:boolean (optional *default*='false')	(No Description)

Child Elements

Table 44: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
ds:Signature	n/a (0..1)	
oval:notes	n/a (0..1)	

< external_variable >

The external_variable element extends the VariableType and defines a variable with some external source. The actual value(s) for the variable is not provided within the OVAL file, but rather it is retrieved during the evaluation of the OVAL Definition from an external source. An unbounded set of possible-value and possible_restriction child elements can be specified that together specify the list of all possible values that an external source is allowed to supply for the external variable. In other words, the value assigned by an external source must match one of the possible_value or possible_restriction elements specified. Each possible_value element contains a single value that could be assigned to the given external_variable while each possible_restriction element outlines a range of possible values. Note that it is not necessary to declare a variable's possible values, but the option is available if desired. If no possible child elements are specified, then the valid values are only bound to the specified datatype of the external variable. Please refer to the description of the PossibleValueType and PossibleRestrictionType complex types for more information.

Extends: oval-def:VariableType

Child Elements

Table 45: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
possible_value	oval-def:PossibleValueType (0..unbounded)	
possible_restriction	oval-def:PossibleRestrictionType (0..unbounded)	

== PossibleValueType ==

The PossibleValueType complex type is used to outline a single expected value of an external variable. The required hint attribute gives a short description of what the value means or represents.

Attributes

Table 46: Attributes

Attribute	Type	Desc.
hint	xsd:string (required)	(No Description)

Simple Content: xsd:anySimpleType

== PossibleRestrictionType ==

The PossibleRestrictionType complex type outlines a range of possible expected value of an external variable. Each possible_restriction element contains an unbounded list of child restriction elements that each specify a range that an actual value may fall in. For example, a restriction element may specify that a value must be less than 10. When multiple restriction elements are present, a valid possible value's evaluation is based on the operator attribute. The operator attribute is set to AND by default. Other valid operation values are explained in the description of the OperatorEnumeration simple type. One can think of the possible_value and possible_restriction elements as an OR'd list of possible values, with the restriction elements as using the selected operation to evaluate its own list of value descriptions. Please refer to the description of the RestrictionType complex type for more information. The required hint attribute gives a short description of what the value means or represents.

Attributes

Table 47: Attributes

Attribute	Type	Desc.
operator	oval:OperatorEnumeration (optional *default*='AND')	(No Description)
hint	xsd:string (required)	(No Description)

Child Elements

Table 48: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
restriction	oval-def:RestrictionType (1..unbounded)	

== RestrictionType ==

The RestrictionType complex type outlines a restriction that is placed on expected values for an external variable. For example, a possible value may be restricted to a integer less than 10. Please refer to the operationEnumeration simple type for a description of the valid operations.

Attributes

Table 49: Attributes

Attribute	Type	Desc.
operation	oval:OperationEnumeration (required)	(No Description)

Simple Content: xsd:anySimpleType

< constant_variable >

The constant_variable element extends the VariableType and defines a variable with a constant value(s). Each constant_variable defines either a single value or a collection of values to be used throughout the evaluation of the OVAL Definition File in which it has been defined. Constant variables cannot be over-ridden by an external source. The actual value of a constant variable is defined by the required value child element. A collection of values can be specified by including multiple instances of the value element. Please refer to the description of the ValueType complex type for more information.

Extends: oval-def:VariableType

Child Elements

Table 50: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
value	oval-def:ValueType (1..unbounded)	

== ValueType ==

The ValueType complex type holds the actual value of the variable when dealing with a constant variable. This value should be used by all tests that reference this variable. The value cannot be over-ridden by an external source.

Simple Content: xsd:anySimpleType

< local_variable >

The local_variable element extends the VariableType and defines a variable with some local source. The actual value(s) for the variable is not provided in the OVAL Definition document but rather it is retrieved during the evaluation of the OVAL Definition. Each local variable is defined by either a single component or a complex function, meaning that a value can be as simple as a literal string or as complex as multiple registry keys concatenated together. Note that if an individual component is used and it returns a collection of values, then there will be multiple values associated with the local_variable. For example, if an object_component is used and it references a file object that identifies a set of 5 files, then the local variable would evaluate to a collection of those 5 values. Please refer to the description of the ComponentGroup for more information.

Extends: oval-def:VariableType

Child Elements

Table 51: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
oval-def:ComponentGroup	n/a (1..1)	

– ComponentGroup –

Any value that is pulled directly off the local system is defined by the basic component element. For example, the name of a user or the value of a registry key. Please refer to the definition of the ObjectComponentType for more information. A value can also be obtained from another variable. The variable element identifies a variable id to pull a value(s) from. Please refer to the definition of the VariableComponentType for more information. Literal values can also be specified.

Child Elements

Table 52: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object_component	oval-def:ObjectComponentType (1..1)	
variable_component	oval-def:VariableComponentType (1..1)	
literal_component	oval-def:LiteralComponentType (1..1)	
oval-def:FunctionGroup	n/a (1..1)	

== LiteralComponentType ==

The LiteralComponentType complex type defines a literal value to be used as a component. The optional datatype attribute defines the type of data expected. The default datatype is 'string'.

Attributes

Table 53: Attributes

Attribute	Type	Desc.
datatype	oval:SimpleDatatypeEnumeration (optional *default*='string')	(No Description)

Simple Content: xsd:anySimpleType

== ObjectComponentType ==

The ObjectComponentType complex type defines a specific value or set of values on the local system to obtain.

The required object_ref attribute provides a reference to an existing OVAL Object declaration. The referenced OVAL Object specifies a set of OVAL Items to collect. Note that an OVAL Object might identify 0, 1, or many OVAL Items on a system. If no items are found on the system then an error should be reported when determining the value of an ObjectComponentType. If 1 or more OVAL Items are found then each OVAL Item will be considered and the ObjectComponentType may have one or more values.

The required item_field attribute specifies the name of the entity whose value will be retrieved from each OVAL Item collected by the referenced OVAL Object. For example, if the object_ref references a win-def:file_object, the item_field may specify the 'version' entity as the field to use as the value of the ObjectComponentType. Note that an OVAL Item may have 0, 1, or many entities whose name matches the specified item_field value. If an entity is not found with a name that matches the value of the item_field an error should be reported when determining the value of an ObjectComponentType. If 1 or more matching entities are found in a single OVAL Item the value of the ObjectComponentType is the list of the values from each of the matching entities.

The optional record_field attribute specifies the name of a field in a record entity in an OVAL Item. The record_field attribute allows the value of a specific field to be retrieved from an entity with a datatype of 'record'. If a field with a matching name attribute value is not found in the referenced OVAL Item entity an error should be reported when determining the value of the ObjectComponentType.

Attributes

Table 54: Attributes

Attribute	Type	Desc.
object_ref	oval:ObjectIDPattern (required)	(No Description)
item_field	oval:NonEmptyStringType (required)	(No Description)
record_field	oval:NonEmptyStringType (optional)	(No Description)

== VariableComponentType ==

The VariableComponentType complex type defines a specific value obtained by looking at the value of another OVAL Variable. The required var_ref attribute provides a reference to the variable. One must make sure that the variable reference does not point to the parent variable that uses this component to avoid a race condition.

Attributes

Table 55: Attributes

Attribute	Type	Desc.
var_ref	oval:VariableIDPattern (required)	(No Description)

– FunctionGroup –

Complex functions have been defined that help determine how to manipulate specific values. These functions can be nested together to form complex statements. Each function is designed to work on a specific type of data. If the data being worked on is not of the correct type, a cast should be attempted before reporting an error. For example, if a concat function includes a registry component that returns an integer, then the integer should be cast as a string in order to work with the concat function. Note that if the operation being applied to the variable by the calling entity is “pattern match”, then all the functions are performed before the regular expression is evaluated. In short, the variable would produce a value as normal and then any pattern match operation would be performed. It is also important to note that when using these functions with sub-components that return a collection of values that the operation will be performed on the Cartesian product of the components and the result is also a collection of values. For example, assume a local_variable specifies the arithmetic function with an arithmetic_operation of “add” and has two sub-components under this function: the first component returns “1” and “2”, and the second component returns “3” and “4” and “5”. The local_variable element would be evaluated to have a collection of six values: 1+3, 1+4, 1+5, 2+3, 2+4, and 2+5. Please refer to the description of a specific function for more details about it.

Child Elements

Table 56: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
arithmetic	oval-def:ArithmeticFunctionType (1..1)	
begin	oval-def:BeginFunctionType (1..1)	
concat	oval-def:ConcatFunctionType (1..1)	
end	oval-def:EndFunctionType (1..1)	
escape_regex	oval-def:EscapeRegexFunctionType (1..1)	
split	oval-def:SplitFunctionType (1..1)	
substring	oval-def:SubstringFunctionType (1..1)	
time_difference	oval-def:TimeDifferenceFunctionType (1..1)	
regex_capture	oval-def:RegexCaptureFunctionType (1..1)	
unique	oval-def:UniqueFunctionType (1..1)	
count	oval-def:CountFunctionType (1..1)	
glob_to_regex	oval-def:GlobToRegexFunctionType (1..1)	

== ArithmeticFunctionType ==

The arithmetic function takes two or more integer or float components and performs a basic mathematical function on them. The result of this function is a single integer or float unless one of the components returns a collection of values. In this case the specified arithmetic function would be performed multiple times and the end result would also be a collection of values for the local variable. For example assume a local_variable specifies the arithmetic function with an arithmetic_operation of “add” and has two sub-components under this function: the first component returns “1” and “2”, and the second component returns “3” and “4” and “5”. The local_variable element would be evaluated to be a collection of six values: 1+3, 1+4, 1+5, 2+3, 2+4, and 2+5.

Note that if both an integer and float components are used then the result is a float.

Attributes

Table 57: Attributes

Attribute	Type	Desc.
arithmetic_operation	oval-def:ArithmeticEnumeration (required)	(No Description)

Child Elements

Table 58: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
oval-def:ComponentGroup	n/a (1..1)	

== BeginFunctionType ==

The begin function takes a single string component and defines a character (or string) that the component string should start with. The character attribute defines the specific character (or string). The character (or string) is only added to the component string if the component string does not already start with the specified character (or string). If the component string does not start with the specified character (or string) the entire character (or string) will be prepended to the component string..

Attributes

Table 59: Attributes

Attribute	Type	Desc.
character	xsd:string (required)	(No Description)

Child Elements

Table 60: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
oval-def:ComponentGroup	n/a (1..1)	

== ConcatFunctionType ==

The concat function takes two or more components and concatenates them together to form a single string. The first component makes up the beginning of the resulting string and any following components are added to the end it. If one of the components returns multiple values then the concat function would be performed multiple times and the end result would be a collection of values for the local variable. For example assume a local variable has two sub-components: a basic component element returns the values “abc” and “def”, and a literal component element that has a value of “xyz”. The local_variable element would evaluate to a collection of two values, “abcxyz” and “defxyz”. If one of the components does not exist, then the result of the concat operation should be does not exist.

Child Elements

Table 61: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
oval-def:ComponentGroup	n/a (1..1)	

Below is a chart that specifies how to classify the flag status of a variable using the concat function during evaluation when multiple components are supplied. Both the object and variable component are indirectly associated with collected objects in a system characteristics file. These objects could have been completely collected from the system, or there might have been some type of error that led to the object not being collected, or maybe only a part of the object set was collected. This flag status is important as OVAL Objects or OVAL States that are working with a variable (through the var_ref attribute on an entity) can use this information to report more accurate results. For example, an OVAL Test with a check attribute of ‘at least one’ that specifies an object with a variable reference, might be able to produce a valid result based on an incomplete object set as long as one of the objects in the set is true. ‘‘‘

|| num of components with flag || || || resulting flag is || E | C | I | DNE | NC | NA ||

——||————————————————||—————— || 1+ | 0+ | 0+ | 0+ | 0+ | 0+ || Error || 0 | 1+ | 0 | 0 | 0 | 0 || Complete || 0 | 0+ | 1+ | 0 | 0 | 0 || Incomplete || 0 | 0+ | 0+ | 1+ | 0 | 0 || Does Not Exist || 0 | 0+ | 0+ | 0+ | 1+ | 0 || Not Collected || 0 | 0+ | 0+ | 0+ | 0+ | 1+ || Not Applicable

——||————————————————||—————— ‘‘‘

== EndFunctionType ==

The end function takes a single string component and defines a character (or string) that the component string should end with. The character attribute defines the specific character (or string). The character (or string) is only added to the component string if the component string does not already end with the specified character (or string). If the desired end character is a string, then the entire end string must exist at the end if the component string. If the entire end string is not present then the entire end string is appended to the component string.

Attributes

Table 62: Attributes

Attribute	Type	Desc.
character	xsd:string (required)	(No Description)

Child Elements

Table 63: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
oval-def:ComponentGroup	n/a (1..1)	

== EscapeRegexFunctionType ==

The escape_regex function takes a single string component and escapes all of the regular expression characters. If the string sub-component contains multiple values, then the escape_regex function will be applied to each individual value and return a multiple-valued result. For example, the string '(.test_string*)?' will evaluate to '(.test_string*)?'. The purpose for this is that many times, a component used in pattern match needs to be treated as a literal string and not a regular expression. For example, assume a basic component element that identifies a file path that is held in the Windows registry. This path is a string that might contain regular expression characters. These characters are likely not intended to be treated as regular expression characters and need to be escaped. This function allows a definition writer to mark convert the values of components to regular expression format.

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. The set of Perl metacharacters which must be escaped by this function is as follows, enclosed by single quotes: '^$.[](){}*+?|'. For more information on the supported regular expression syntax in OVAL see: http://oval.mitre.org/language/about/re_support_5.6.html.

Child Elements

Table 64: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
oval-def:ComponentGroup	n/a (1..1)	

== SplitFunctionType ==

The split function takes a single string component and turns it into a collection of values based on a delimiter string. For example, assume that a basic component element returns the value “a-b-c-d” to the split function with the delimiter set to “-”. The local_variable element would be evaluated to have four values “a”, “b”, “c”, and “d”. If the basic component returns a value that begins, or ends, with a delimiter, the local_variable element would contain empty string values at the beginning, or end, of the collection of values returned for that string component. For example, if the delimiter is “-”, and the basic component element returns the value “-a-a-”, the local_variable element would evaluate to a collection of four values “”, “a”, “a”, and “”. Likewise, if the basic component element returns a value that contains adjacent delimiters such as “—”, the local_variable element would evaluate to a collection of four values “”, “”, “”, and “”. Lastly, if the basic component element used by the split function returnsa collection of values, then the split function is performed multiple times, and all of the results, from each of the split functions, are returned.

Attributes

Table 65: Attributes

Attribute	Type	Desc.
delimiter	xsd:string (required)	(No Description)

Child Elements

Table 66: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
oval-def:ComponentGroup	n/a (1..1)	

== SubstringFunctionType ==

The substring function takes a single string component and produces a single value that contains a portion of the original string. The substring_start attribute defines the starting position in the original string. To include the first character of the string, the start position would be 1. A value less than 1 also means that the start position would be 1. If the substring_start attribute has value greater than the length of the original string an error should be reported. The substring_length attribute defines how many characters after, and including, the starting character to include. A substring_length value greater than the actual length of the string, or a negative value, means to include all of the characters after the starting character. For example, assume a basic component element that returns the value “abcdefg” with a substring_start value of 3 and a substring_length value of 2. The local_variable element would evaluate to have a single value of “cd”. If the string component used by the substring function returns a collection of values, then the substring operation is performed multiple times and results in a collection of values for the component.

Attributes

Table 67: Attributes

Attribute	Type	Desc.
substring_start	xsd:int (required)	(No Description)
substring_length	xsd:int (required)	(No Description)

Child Elements

Table 68: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
oval-def:ComponentGroup	n/a (1..1)	

== TimeDifferenceFunctionType ==

The time_difference function calculates the difference in seconds between date-time values. If one component is specified, the values of that component are subtracted from the current time (UTC). The current time is the time at which the function is evaluated. If two components are specified, the value of the second component is subtracted from the value of the first component. If the component(s) contain a collection of values, the operation is performed multiple times on the Cartesian product of the component(s) and the result is also a collection of time difference values. For example, assume a local_variable specifies the time_difference function and has two sub-components under this function: the first component returns “04/02/2009” and “04/03/2009”, and the second component returns “02/02/2005” and “02/03/2005” and “02/04/2005”. The local_variable element would evaluate to a collection of six values: (ToSeconds(“04/02/2009”) - ToSeconds(“02/02/2005”)), (ToSeconds(“04/02/2009”) - ToSeconds(“02/03/2005”)), (ToSeconds(“04/02/2009”) - ToSeconds(“02/04/2005”)), (ToSeconds(“04/03/2009”) - ToSeconds(“02/02/2005”)), (ToSeconds(“04/03/2009”) - ToSeconds(“02/03/2005”)), and (ToSeconds(“04/03/2009”) - ToSeconds(“02/04/2005”)).

The date-time format of each component is determined by the two format attributes. The format1 attribute applies to the first component, and the format2 attribute applies to the second component. Valid values for the attributes are ‘win_filetime’, ‘seconds_since_epoch’, ‘day_month_year’, ‘year_month_day’, and ‘month_day_year’. Please see the DateTimeFormatEnumeration for more information about each of these values. If an input value is not understood, the result is an error. If only one input is specified, specify the format with the format2 attribute, as the first input is considered to be the implied ‘current time’ input.

Note that the datatype associated with the components should be ‘string’ or ‘int’ depending on which date time format is specified. The result of this function though is always an integer.

Attributes

Table 69: Attributes

Attribute	Type	Desc.
format_1	oval-def:DateTimeFormatEnumeration (optional *default*='year_month_day')	(No Description)
format_2	oval-def:DateTimeFormatEnumeration (optional *default*='year_month_day')	(No Description)

Child Elements

Table 70: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
oval-def:ComponentGroup	n/a (1..1)	

== RegexCaptureFunctionType ==

The regex_capture function captures a single substring from a single string component. If the string sub-component contains multiple values, then the regex_capture function will extract a substring from each value. The ‘pattern’ attribute provides a regular expression that should contain a single subexpression (using parentheses). For example, the pattern ^abc(.*)xyz$ would capture a substring from each of the string component's values if the value starts with abc and ends with xyz. In this case the subexpression would be all the characters that exist in between the abc and the xyz. Note that subexpressions match the longest possible substrings.

If the regular expression contains multiple capturing sub-patterns, only the first capture is used. If there are no capturing sub-patterns, the result for each target string must be the empty string. Otherwise, if the regular expression could match the target string in more than one place, only the first match (and its first capture) is used. If no matches are found in a target string, the result for that target must be the empty string.

Note that a quantified capturing sub-pattern does not produce multiple substrings. Standard regular expression semantics are such that if a capturing sub-pattern is required to match multiple times in order for the overall regular expression to match, the capture produced is the last substring to have matched the sub-pattern.

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. If any of the Perl metacharacters are to be used literally, then they must be escaped. The set of metacharacters which must be escaped for this purpose is as follows, enclosed by single quotes: '^$.[](){}*+?|'. For more information on the supported regular expression syntax in OVAL see: http://oval.mitre.org/language/about/re_support_5.6.html.

Attributes

Table 71: Attributes

Attribute	Type	Desc.
pattern	xsd:string	(No Description)

Child Elements

Table 72: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
oval-def:ComponentGroup	n/a (1..1)	

== UniqueFunctionType ==

The unique function takes one or more components and removes any duplicate value from the set of components. All components used in the unique function will be treated as strings. For example, assume that three components exist, one that contains a string value of 'foo', and two of which both resolve to the string value 'bar'. Applying the unique function to these three components resolves to a local_variable with two string values, 'foo' and 'bar'. Additionally, if any of the components referenced by the unique function evaluate to a collection of values, then those values are used in the unique calculation. For example, assume that there are two components, one of which resolves to a single string value, 'foo', the other of which resolves to two string values, 'foo' and 'bar'. If the unique function is used to remove duplicates from these two components, the function will resolve to a local_variable that is a collection of two string values, 'foo' and 'bar'.

Child Elements

Table 73: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
oval-def:ComponentGroup	n/a (1..1)	

== CountFunctionType ==

The count function takes one or more components and returns the count of all of the values represented by the components. For example, assume that two variables exist, each with a single value. By applying the count function against two variable components that resolve to the two variables, the resulting local_variable would have a value of '2'. Additionally, if any of the components referenced by the count function evaluate to a collection of values, then those values are used in the count calculation. For example, assume that there are two components, one of which resolves to a single value, the other of which resolves to two values. If the count function is used to provide a count of these two components, the function will resolve to a local_variable with the values '3'.

Child Elements

Table 74: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
oval-def:ComponentGroup	n/a (1..1)	

== GlobToRegexFunctionType ==

The glob_to_regex function takes a single string component representing shell glob pattern and produces a single value that corresponds to result of a conversion of the original glob pattern into Perl 5's regular expression pattern. The glob_noescape attribute defines the way how the backslash ('') character should be interpreted. It defaults to 'false' meaning backslash should be interpreted as an escape character (backslash is allowed to be used as an escape character). If the glob_noescape attribute would be set to 'true' it instructs the glob_to_regex function to interpret the backslash ('') character as a literal, rather than as an escape character (backslash is *not* allowed to be used as an escape character). Refer to table with examples below to see the difference how a different boolean value of the 'glob_noescape' attribute will impact the output form of the resulting Perl 5's regular expression produced by glob_to_regex function.

Please note the glob_to_regex function will fail to perform the conversion and return an error when the provided string argument (to represent glob pattern) does not represent a syntactically correct glob pattern. For example given the 'a*b?[' as the argument to be converted, glob_to_regex would return an error since there's missing the corresponding closing bracket in the provided glob pattern argument.

Also, it is necessary to mention that the glob_to_regex function respects the default behaviour for the input glob pattern and output Perl 5's regular expression spaces. Namely this means that:

- glob_to_regex will respect the UNIX glob behavior when processing forward slashes, forward slash should be treated as a path separator and * or ? shall not match it,
- glob_to_regex will rule out matches having special meaning (for example '.' as a representation of the current working directory or '..' as a representation of the parent directory of the current working directory,
- glob_to_regex will rule out files or folders starting with '.' character (e.g. dotfiles) unless the respective glob pattern part itself starts with the '.' character,

- glob_to_regex will not perform case-sensitivity transformation (alphabetical characters will be copied from input glob pattern space to output Perl 5’s regular expression pattern space intact). It is kept as a responsibility of the OVAL content author to provide input glob pattern argument in such case so the resulting Perl 5’s regular expression pattern will match the expected pathname entries according to the case of preference,
- glob_to_regex will not perform any possible brace expansion. Therefore glob patterns like ‘{pat,pat,pat}’ would be converted into Perl 5’s regular expression syntax in the original un-expanded form (kept for any potential subsequent expansion to be performed by Perl 5’s regular expression engine in the moment of the use of that resulting regular expression),
- glob_to_regex will not perform tilde (‘~’) character substitution to user name home directory pathname. The (‘~’) character will be passed to Perl 5’s regular expression engine intact. If user name home directory pathname glob pattern behaviour is expected, the pathname of the user name home directory needs to be specified in the original input glob pattern already,
- glob_to_regex function will not perform any custom changes wrt to the ordering of items (perform any additional sorting of set of pathnames represented by the provided glob pattern argument).

Attributes

Table 75: Attributes

Attribute	Type	Desc.
glob_noescape	xsd:boolean (optional *default*=’false’)	(No Description)

Child Elements

Table 76: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
oval-def:ComponentGroup	n/a (1..1)	

Below are some examples that outline how the glob_noescape attribute value affects the output form of the produced Perl regular expression. The far left column identifies the shell glob pattern provided as the input string component to the glob_to_regex function. The middle column specifies the two possible different boolean values of the ‘glob_noescape’ attribute that can be used. Finally the last column depicts how the output produced by the glob_to_regex function - the resulting Perl regular expression would look like. ‘‘‘

|| ||

input shell glob pattern || glob_noescape attribute value || corresponding Perl regular expression ||
||

——————————||————————————-||——————————————

‘*’ || false || ^*$ ||————————————-||——————————————

‘*’ || true || ^\[^/]*$

——————————||————————————-||——————————————

‘?’ || false || ^?$ ||————————————-||——————————————

‘?’ || true || ^\[^./]$

——————————||————————————-||——————————————

‘[hello]’ || false || ^[hello]$ ||————————————-||——————————————

‘[hello]’ || true || ^\[hello\]$

————————————||———————————————-||——————————————————

‘/root/*’ || false || ^/root/(?=[^.])[^/]*$ ||———————————————-||——————————————————

‘/root/.*’ || false || ^/root/.[^/]*$ ||———————————————-||——————————————————

‘/root/x*’ || false || ^/root/x[^/]*$ ||———————————————-||——————————————————

‘/root/?’ || false || ^/root/[^./]$ ||———————————————-||——————————————————

‘/root/.?’ || false || ^/root/.[^/]$ ||———————————————-||——————————————————

‘/root/x?’ || false || ^/root/x[^/]$

————————————||———————————————-||——————————————————

‘list.?’ || false || ^list.[^/]$ ||———————————————-||——————————————————

‘list.?’ || true || ^list.[^/]$ ||———————————————-||——————————————————

‘project.*’ || false || ^project.[^/]*$ ||———————————————-||——————————————————

‘project.*’ || true || ^project.[^/]*$ ||———————————————-||——————————————————

‘*old’ || false || ^(?=[^.])[^/]*old$ ||———————————————-||——————————————————

‘*old’ || true || ^(?=[^.])[^/]*old$ ||———————————————-||——————————————————

‘type*.[ch]’ || false || ^type[^/]*.[ch]$ ||———————————————-||——————————————————

‘type*.[ch]’ || true || ^type[^/]*.[ch]$ ||———————————————-||——————————————————

‘.’ || false || ^(?=[^.])[^/]*.[^/]*$ ||———————————————-||——————————————————

‘.’ || true || ^(?=[^.])[^/]*.[^/]*$

||———————————————-||——————————————————

‘*’ || false || ^(?=[^.])[^/]*$ ||———————————————-||——————————————————

‘*’ || true || ^(?=[^.])[^/]*$ ||———————————————-||——————————————————

‘?’ || false || ^[^./]$ ||———————————————-||——————————————————

‘?’ || true || ^[^./]$ ||———————————————-||——————————————————

‘*’ || false || ^*$ ||———————————————-||——————————————————

‘*’ || true || ^\[^/]*$ ||———————————————-||——————————————————

‘?’ || false || ^?$ ||———————————————-||——————————————————

‘?’ || true || ^\[^./]$ ||———————————————-||——————————————————

‘x[[:digit:]]*’ || false || ^x[[:digit:]]*$ ||———————————————-||——————————————————

‘x[[:digit:]]*’ || true || ^x[[:digit:]]\[^/]*$

||———————————————-||——————————————————

‘’ || false || ^$ ||———————————————-||——————————————————

‘’ || true || ^$ ||———————————————-||——————————————————

‘~/files/*.txt’* || *false* || ^~*/files/(?=[^.])[^/]*.txt$ ||———————————————-||——————————————————

‘~/files/*.txt’* || *true* || ^~*/files/(?=[^.])[^/]*.txt$

||———————————————-||——————————————————

‘’ || false || ^\$ ||————————————-||————————————

‘’ || true || ^\$ ||————————————-||————————————

‘[ab’ || false || INVALID ||————————————-||————————————

‘[ab’ || true || INVALID ||————————————-||————————————

‘.*.conf’ || false || ^.[^/]*.conf$ ||————————————-||————————————

‘.*.conf’ || true || ^.[^/]*.conf$ ||————————————-||————————————

‘docs/?b’ || false || ^docs/[^./]b$ ||————————————-||————————————

‘docs/?b’ || true || ^docs/[^./]b$ ||————————————-||————————————

‘xy/??z’ || false || ^xy/[^./][^/]z$ ||————————————-||————————————

‘xy/??z’ || true || ^xy/[^./][^/]z$

– ArithmeticEnumeration –

The ArithmeticEnumeration simple type defines basic arithmetic operations. Currently add and multiply are defined.

Table 77: Enumeration Values

Value	Description
add	(No Description)
multiply	(No Description)

– DateTimeFormatEnumeration –

The DateTimeFormatEnumeration simple type defines the different date-time formats that are understood by OVAL. Note that in some cases there are a few different possibilities within a given format. Each of these possibilities is unique though and can be distinguished from each other. The different formats are used to clarify the higher level structure of the date-time string being used.

Table 78: Enumeration Values

Value	Description
year_month_day	The year_month_day value specifies date-time strings that follow the formats: ‘yyyymmdd’, ‘yyyymmddThhmmss’, ‘yyyy/mm/dd hh:mm:ss’, ‘yyyy/mm/dd’, ‘yyyy-mm-dd hh:mm:ss’, or ‘yyyy-mm-dd’
month_day_year	The month_day_year value specifies date-time strings that follow the formats: ‘mm/dd/yyyy hh:mm:ss’, ‘mm/dd/yyyy’, ‘mm-dd-yyyy hh:mm:ss’, ‘mm-dd-yyyy’, ‘NameOfMonth, dd yyyy hh:mm:ss’ or ‘NameOfMonth, dd yyyy’, ‘AbreviatedNameOfMonth, dd yyyy hh:mm:ss’, or ‘AbreviatedNameOfMonth, dd yyyy’
day_month_year	The day_month_year value specifies date-time strings that follow the formats: ‘dd/mm/yyyy hh:mm:ss’, ‘dd/mm/yyyy’, ‘dd-mm-yyyy hh:mm:ss’, or ‘dd-mm-yyyy’
win_filetime	The win_filetime value specifies date-time strings that follow the windows file time format.
seconds_since_epoch	The seconds_since_epoch value specifies date-time values that represent the time in seconds since the UNIX epoch. The Unix epoch is the time 00:00:00 UTC on January 1, 1970.
cim_datetime	The cim_datetime model is used by WMI and its value specifies date-time strings that follow the format: ‘yyyymmddHHMMSS.mmmmmmsUUU’, and alternatively ‘yyyy-mm-dd HH:MM:SS:mmm’ only when used in WMI Query Language queries.

– FilterActionEnumeration –

The FilterActionEnumeration simple type defines the different options for filtering sets of items.

Table 79: Enumeration Values

Value	Description
exclude	The exclude value specifies that all items that match the filter shall be excluded from set that the filter is applied to.
include	The include value specifies that only items that match the filter shall be included in the set that the filter is applied to.

– SetOperatorEnumeration –

The SetOperatorEnumeration simple type defines acceptable set operations. Set operations are used to take multiple different sets of objects within OVAL and merge them into a single unique set. The different operators that guide this merge are defined below. For each operator, if only a single object has been supplied, then the resulting set is simply that complete object.

Table 80: Enumeration Values

Value	Description
COMPLEMENT	The complement operator is defined in OVAL as a relative complement. The resulting unique set contains everything that belongs to the first declared set that is not part of the second declared set. If A and B are sets (with A being the first declared set), then the relative complement is the set of elements in A, but not in B, with the duplicates removed.
INTERSECTION	The intersection of two sets in OVAL results in a unique set that contains everything that belongs to both sets in the collection, but nothing else. If A and B are sets, then the intersection of A and B contains all the elements of A that also belong to B, but no other elements, with the duplicates removed.
UNION	The union of two sets in OVAL results in a unique set that contains everything that belongs to either of the original sets. If A and B are sets, then the union of A and B contains all the elements of A and all elements of B, with the duplicates removed.

Below are some tables that outline how different flags are combined with a given set_operator to return a new flag.

These tables are needed when computing the flag for collected objects that represent object sets in an OVAL Definition. The top row identifies the flag associated with the first set or object reference. The left column identifies the flag associated with the second set or object reference. The matrix inside the table represent the resulting flag when the given set_operator is applied. (E=error, C=complete, I=incomplete, DNE=does not exist, NC=not collected, NA=not applicable) ‘‘‘

|| ||

set_operator is || obj 1 flag ||

union || || || E | C | I | DNE | NC | NA ||

——————————||————————————————————||

E || E | E | E | E | E | E ||

obj C || E | C | I | C | I | C || 2 I || E | I | I | I | I | I ||

flag DNE || E | C | I | DNE | I | DNE || NC || E | I | I | I | NC | NC || NA || E | C | I | DNE | NC | NA ||

——————————||————————————————————|| ‘‘‘

‘‘‘

|| ||

set_operator is || obj 1 flag ||

intersection || || || E | C | I | DNE | NC | NA ||

——————————||————————————————————||

E || E | E | E | DNE | E | E ||

obj C || E | C | I | DNE | NC | C || 2 I || E | I | I | DNE | NC | I ||

flag DNE || DNE | DNE | DNE | DNE | DNE | DNE || NC || E | NC | NC | DNE | NC | NC || NA || E | C | I | DNE | NC | NA ||

——————————||————————————————————|| ‘‘‘

‘‘‘

|| ||

set_operator is || obj 1 flag ||

complement || || || E | C | I | DNE | NC | NA ||

——————————||————————————————————||

E || E | E | E | DNE | E | E ||

obj C || E | C | I | DNE | NC | E || 2 I || E | E | E | DNE | NC | E ||

flag DNE || E | C | I | DNE | NC | E || NC || E | NC | NC | DNE | NC | E || NA || E | E | E | E | E | E ||

——————————||————————————————————|| ‘‘‘

– EntityAttributeGroup –

The EntityAttributeGroup is a collection of attributes that are common to all entities. This group defines these attributes and their default values. Individual entities may limit allowed values for these attributes, but all entities will support these attributes.

Attributes

Table 81: Attributes

Attribute	Type	Desc.
datatype	oval:DatatypeEnumeration (optional *default*=’string’)	The optional datatype attribute specifies how the given operation should be applied to the data. Since we are dealing with XML everything is technically a string, but often the value is meant to represent some other datatype and this affects the way an operation is performed. For example, with the statement ‘is 123 less than 98’. If the data is treated as integers the answer is no, but if the data is treated as strings, then the answer is yes. Specifying a datatype defines how the less than operation should be performed. Another way of thinking of things is that the datatype attribute specifies how the data should be cast before performing the operation (note that the default datatype is ‘string’). In the previous example, if the datatype is set to int, then ‘123’ and ‘98’ should be cast as integers. Another example is applying the ‘equals’ operation to ‘1.0.0.0’ and ‘1.0’. With datatype ‘string’ they are not equal, with datatype ‘version’ they are. Note that there are certain cases where a cast from one datatype to another is not possible. If a cast cannot be made, (trying to cast ‘abc’ to an integer) then an error should be reported. For example, if the datatype is set to ‘integer’ and the value is the empty string. There is no way to cast the empty string (or NULL) to an integer, and in cases like this an error should be reported.
operation	oval:OperationEnumeration (optional *default*=’equals’)	The optional operation attribute determines how the individual entities should be evaluated (the default operation is ‘equals’).
mask	xsd:boolean (optional *default*=’false’)	The optional mask attribute is used to identify values that have been hidden for sensitivity concerns. This is used by the Result document which uses the System Characteristics schema to format the information found on a specific system. When the mask attribute is set to ‘true’ on an OVAL Entity or an OVAL Field, the corresponding collected value of that OVAL Entity or OVAL Field MUST NOT be present in the “results” section of the OVAL Results document; the “oval_definitions” section must not be altered and must be an exact copy of the definitions evaluated. Values MUST NOT be masked in OVAL System Characteristics documents that are not contained within an OVAL Results document. It is possible for masking conflicts to occur where one entity has mask set to true and another entity has mask set to false. A conflict will occur when the mask attribute is set differently on an OVAL Object and matching OVAL State or when more than one OVAL Objects identify the same OVAL Item(s). When such a conflict occurs the result is always to mask the entity.
var_ref	oval:VariableIDPattern (optional)	The optional var_ref attribute refers the value of the element to a variable element. When supplied, the value(s) associated with the OVAL Variable should be used as the value(s) of the element. If there is an error computing the value of the variable, then that error should be passed up to the element referencing it. If the variable being referenced does not have a value (for example, if the variable pertains to the size of a file, but the file does not exist) then one of two results are possible. If the element is part of an object declaration, then the object element referencing it is considered to not exist. If the element is part of a state declaration, then the state element referencing it will evaluate to error.
var_check	oval:CheckEnumeration (optional)	The optional var_check attribute specifies how data collection or state evaluation should proceed when an element uses a var_ref attribute, and the associated variable defines more than one value. For example, if an object entity ‘filename’ with an operation of ‘not equal’ references a variable that returns five different values, and the var_check attribute has a value of ‘all’, then an actual file on the system matches only if the actual filename does not equal any of the variable values. As another example, if a state entity ‘size’ with an operation of ‘less than’ references a variable that has five different integer values, and the var_check attribute has a value of ‘all’, then the ‘size’ state entity evaluates to true only if the corresponding ‘size’ item entity is less than each of the five integers defined by the variable. If a variable does not have any value value when referenced by an OVAL Object the object should be considered to not exist. If a variable does not have any value when referenced by an OVAL State an error should be reported during OVAL analysis. When an OVAL State uses a var_ref, if both the state entity and a corresponding item entity are collections of values, the var_check is applied to each value of the item entity individually, and all must evaluate to true for the state entity to evaluate to true. In this condition, there is no value of var_check which enables an element-wise comparison, and so there is no way to determine whether the two entities are truly ‘equal’ in that sense. If var_ref is present but var_check is not, the element should be processed as if

== EntitySimpleBaseType ==

The EntitySimpleBaseType complex type is an abstract type that defines the default attributes associated with every simple entity. Entities can be found in both OVAL Objects and OVAL States and represent the individual properties associated with items found on a system. An example of a single entity would be the path of a file. Another example would be the version of the file.

Simple Content: xsd:anySimpleType

== EntityComplexBaseType ==

The EntityComplexBaseType complex type is an abstract type that defines the default attributes associated with every complex entity. Entities can be found in both OVAL Objects and OVAL States and represent the individual properties associated with items found on a system. An example of a single entity would be the path of a file. Another example would be the version of the file.

== EntityObjectIPAddressType ==

The EntityObjectIPAddressType type is extended by the entities of an individual OVAL Object. This type provides uniformity to each object entity by including the attributes found in the EntitySimpleBaseType. This specific type describes any IPv4/IPv6 address or address prefix.

Restricts: oval-def:EntitySimpleBaseType

Attributes

Table 82: Attributes

Attribute	Type	Desc.
datatype	Restriction of oval:SimpleDatatypeEnumeration (required) (‘ipv4_address’, ‘ipv6_address’)	(No Description)

Simple Content: Restricts xsd:string

== EntityObjectIPAddressStringType ==

The EntityObjectIPAddressStringType type is extended by the entities of an individual OVAL Object. This type provides uniformity to each object entity by including the attributes found in the EntitySimpleBaseType. This specific type describes any IPv4/IPv6 address, address prefix, or its string representation.

Restricts: oval-def:EntitySimpleBaseType

Attributes

Table 83: Attributes

Attribute	Type	Desc.
datatype	Restriction of oval:SimpleDatatypeEnumeration (optional *default*='string') ('ipv4_address', 'ipv6_address', 'string')	(No Description)

Simple Content: Restricts xsd:string

== EntityObjectAnySimpleType ==

The EntityObjectAnySimpleType type is extended by the entities of an individual OVAL Object. This type provides uniformity to each object entity by including the attributes found in the EntitySimpleBaseType. This specific type describes any simple data.

Restricts: oval-def:EntitySimpleBaseType

Attributes

Table 84: Attributes

Attribute	Type	Desc.
datatype	oval:SimpleDatatypeEnumeration (optional *default*='string')	(No Description)

Simple Content: Restricts xsd:string

== EntityObjectBinaryType ==

The EntityBinaryType type is extended by the entities of an individual OVAL Object. This type provides uniformity to each object entity by including the attributes found in the EntitySimpleBaseType. This specific type describes simple binary data. The empty string is also allowed when using a variable reference with an element.

Restricts: oval-def:EntitySimpleBaseType

Attributes

Table 85: Attributes

Attribute	Type	Desc.
datatype	oval:SimpleDatatypeEnumeration (required *fixed*='binary')	(No Description)

Simple Content: Union of xsd:hexBinary, oval:EmptyStringType

== EntityObjectBoolType ==

The EntityBoolType type is extended by the entities of an individual OVAL Object. This type provides uniformity to each object entity by including the attributes found in the EntitySimpleBaseType. This specific type describes simple boolean data. The empty string is also allowed when using a variable reference with an element.

Restricts: oval-def:EntitySimpleBaseType

Attributes

Table 86: Attributes

Attribute	Type	Desc.
datatype	oval:SimpleDatatypeEnumeration (required *fixed*='boolean')	(No Description)

Simple Content: Union of xsd:boolean, oval:EmptyStringType

== EntityObjectFloatType ==

The EntityObjectFloatType type is extended by the entities of an individual OVAL Object. This type provides uniformity to each object entity by including the attributes found in the EntitySimpleBaseType. This specific type describes simple float data. The empty string is also allowed when using a variable reference with an element.

Restricts: oval-def:EntitySimpleBaseType

Attributes

Table 87: Attributes

Attribute	Type	Desc.
datatype	oval:SimpleDatatypeEnumeration (required *fixed*='float')	(No Description)

Simple Content: Union of xsd:float, oval:EmptyStringType

== EntityObjectIntType ==

The EntityIntType type is extended by the entities of an individual OVAL Object. This type provides uniformity to each object entity by including the attributes found in the EntitySimpleBaseType. This specific type describes simple integer data. The empty string is also allowed when using a variable reference with an element.

Restricts: oval-def:EntitySimpleBaseType

Attributes

Table 88: Attributes

Attribute	Type	Desc.
datatype	oval:SimpleDatatypeEnumeration (required *fixed*='int')	(No Description)

Simple Content: Union of xsd:integer, oval:EmptyStringType

== EntityObjectStringType ==

The EntityStringType type is extended by the entities of an individual OVAL Object. This type provides uniformity to each object entity by including the attributes found in the EntitySimpleBaseType. This specific type describes simple string data.

Restricts: oval-def:EntitySimpleBaseType

Attributes

Table 89: Attributes

Attribute	Type	Desc.
datatype	oval:SimpleDatatypeEnumeration (optional *fixed*='string')	(No Description)

Simple Content: Restricts xsd:string

== EntityObjectVersionType ==

The EntityObjectVersionType type is extended by the entities of an individual OVAL State. This type provides uniformity to each state entity by including the attributes found in the EntityStateSimpleBaseType. This specific type describes simple version data.

Restricts: oval-def:EntitySimpleBaseType

Attributes

Table 90: Attributes

Attribute	Type	Desc.
datatype	oval:SimpleDatatypeEnumeration (required *fixed*='version')	(No Description)

Simple Content: Restricts xsd:string

== EntityObjectRecordType ==

The EntityObjectRecordType defines an entity that consists of a number of uniquely named fields. This structure is used for representing a record from a database query and other similar structures where multiple related fields must be represented at once. Note that for all entities of this type, the only allowed datatype is 'record' and the only allowed operation is 'equals'. During analysis of a system characteristics item, each field is analyzed and then the overall result for elements of this type is computed by logically anding the results for each field and then applying the entity_check attribute.

Note the datatype attribute must be set to 'record'.

Note the operation attribute must be set to 'equals'.

Note the var_ref attribute is not permitted and the var_check attribute does not apply.

Note that when the mask attribute is set to 'true', all child field elements must be masked regardless of the child field's mask attribute value.

Extends: oval-def:EntityComplexBaseType

Child Elements

Table 91: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
field	oval-def:EntityObjectFieldType (0..unbounded)	

== EntityObjectFieldType ==

The EntityObjectFieldType defines an element with simple content that represents a named field in a record that may contain any number of named fields. The EntityObjectFieldType is much like all other entities with one significant difference, the EntityObjectFieldType has a name attribute

The required name attribute specifies a unique name for the field. Field names are lowercase and must be unique within a given parent record element. When analyzing system characteristics an error should be reported for the result of a field that is present in the OVAL State, but not found in the system characteristics Item.

The optional entity_check attribute specifies how to handle multiple record fields with the same name in the OVAL Systems Characteristics file. For example, while collecting group information where one field is the represents the users that are members of the group. It is very likely that there will be multiple fields with a name of ‘user’ associated with the group. If the OVAL State defines the value of the field with name equal ‘user’ to equal ‘Fred’, then the entity_check attribute determines if all values for field entities must be equal to ‘Fred’, or at least one value must be equal to ‘Fred’, etc.

Note that when the mask attribute is set to ‘true’ on a field’s parent element the field must be masked regardless of the field’s mask attribute value.

Attributes

Table 92: Attributes

Attribute	Type	Desc.
name	Restriction of xsd:string (required)	A string restricted to disallow upper case characters.
entity_check	oval:CheckEnumeration (optional *default*=’all’)	(No Description)

Simple Content: xsd:anySimpleType

== EntityStateSimpleBaseType ==

The EntityStateSimpleBaseType complex type is an abstract type that extends the EntitySimpleBaseType and is used by some entities within an OVAL State.

The optional check_existence attribute specifies how to interpret the status of corresponding item entities when performing an item-state comparison. The default value for this attribute is ‘at_least_one_exists’ indicating that by default an item comparison may evaluate to true only if at least one corresponding item entity has a status of ‘exists’. For example, if a value of ‘none_exist’ is given, then the comparison can evaluate to true only if there are one or more corresponding item entities, each with a status of ‘does not exist’.

The optional entity_check attribute specifies how to handle multiple item entities with the same name in the OVAL Systems Characteristics file. For example, suppose we are dealing with a Group Test and an entity in the state is related to the user. It is very likely that when the information about the group is collected off of the system (and represented in the OVAL System Characteristics file) that there will be multiple users associated with the group (i.e. multiple ‘user’ item entities associated with the same ‘user’ state entity). If the OVAL State defines the value of the user entity to equal ‘Fred’, then the entity_check attribute determines if all values for ‘user’ item entities must be equal to ‘Fred’, or at least one value must be equal to ‘Fred’, etc. Note that with the exception of the ‘none_satisfy’ check value, the entity_check attribute can only affect the result of the test if the corresponding OVAL Item allows more than one occurrence of the entity (e.g. ‘maxOccurs’ is some value greater than one).

The entity_check and var_check attributes are considered together when evaluating a single state entity. When a variable identifies more than one value and multiple item entities with the same name exist, for a single state entity, a many-to-many comparison must be conducted. In this situation, there are many values for the state entity that must be compared to many item entities. Each item entity is compared to the state entity. For each item entity, an interim result is calculated by using the var_check attribute to combine the result of comparing each variable value with a single system value. Then these interim results are combined for each system value using the entity_check attribute.

Attributes

Table 93: Attributes

Attribute	Type	Desc.
entity_check	oval:CheckEnumeration (optional *default*=’all’)	(No Description)
check_existence	oval:ExistenceEnumeration (optional *default*=’at_least_one_exists’)	(No Description)

Simple Content: oval-def:EntitySimpleBaseType

== EntityStateComplexBaseType ==

The EntityStateComplexBaseType complex type is an abstract type that extends the EntityComplexBaseType and is used by some entities within an OVAL State.

The optional check_existence attribute specifies how to interpret the status of corresponding item entities when performing an item-state comparison. The default value for this attribute is ‘at_least_one_exists’ indicating that by default an item comparison may evaluate to true only if at least one corresponding item entity has a status of ‘exists’. For example, if a value of ‘none_exist’ is given, then the comparison can evaluate to true only if there are one or more corresponding item entities, each with a status of ‘does not exist’.

The optional entity_check attribute specifies how to handle multiple item entities with the same name in the OVAL Systems Characteristics file. For example, suppose we are dealing with a Group Test and an entity in the state is related to the user. It is very likely that when the information about the group is collected off of the system (and represented in the OVAL System Characteristics file) that there will be multiple users associated with the group (i.e. multiple ‘user’ item entities associated with the same ‘user’ state entity). If the OVAL State defines the value of the user entity to equal ‘Fred’, then the entity_check attribute determines if all values for ‘user’ item entities must be equal to ‘Fred’, or at least one value must be equal to ‘Fred’, etc. Note that with the exception of the ‘none_satisfy’ check value, the entity_check attribute can only affect the result of the test if the corresponding OVAL Item allows more than one occurrence of the entity (e.g. ‘maxOccurs’ is some value greater than one).

The entity_check and var_check attributes are considered together when evaluating a single state entity. When a variable identifies more than one value and multiple item entities with the same name exist, for a single state entity, a many-to-many comparison must be conducted. In this situation, there are many values for the state entity that must be compared to many item entities. Each item entity is compared to the state entity. For each item entity, an interim result is calculated by using the var_check attribute to combine the result of comparing each variable value with a single system value. Then these interim results are combined for each system value using the entity_check attribute.

Extends: oval-def:EntityComplexBaseType

Attributes

Table 94: Attributes

Attribute	Type	Desc.
entity_check	oval:CheckEnumeration (optional *default*='all')	(No Description)
check_existence	oval:ExistenceEnumeration (optional *default*='at_least_one_exists')	(No Description)

== EntityStateIPAddressType ==

The EntityStateIPAddressType type is extended by the entities of an individual OVAL State. This type provides uniformity to each object entity by including the attributes found in the EntityStateSimpleBaseType. This specific type describes any IPv4/IPv6 address or address prefix.

Restricts: oval-def:EntityStateSimpleBaseType

Attributes

Table 95: Attributes

Attribute	Type	Desc.
datatype	Restriction of oval:SimpleDatatypeEnumeration (required) ('ipv4_address', 'ipv6_address')	(No Description)

Simple Content: Restricts xsd:string

== EntityStateIPAddressStringType ==

The EntityStateIPAddressStringType type is extended by the entities of an individual OVAL State. This type provides uniformity to each object entity by including the attributes found in the EntityStateSimpleBaseType. This specific type describes any IPv4/IPv6 address, address prefix, or its string representation.

Restricts: oval-def:EntityStateSimpleBaseType

Attributes

Table 96: Attributes

Attribute	Type	Desc.
datatype	Restriction of oval:SimpleDatatypeEnumeration (optional *default*='string') ('ipv4_address', 'ipv6_address', 'string')	(No Description)

Simple Content: Restricts xsd:string

== EntityStateAnySimpleType ==

The EntityStateAnySimpleType type is extended by the entities of an individual OVAL State. This type provides uniformity to each state entity by including the attributes found in the EntityStateSimpleBaseType. This specific type describes any simple data.

Restricts: oval-def:EntityStateSimpleBaseType

Attributes

Table 97: Attributes

Attribute	Type	Desc.
datatype	oval:SimpleDatatypeEnumeration (optional *default*='string')	(No Description)

Simple Content: Restricts xsd:string

== EntityStateBinaryType ==

The EntityStateBinaryType type is extended by the entities of an individual OVAL State. This type provides uniformity to each state entity by including the attributes found in the EntityStateSimpleBaseType. This specific type describes simple binary data. The empty string is also allowed when using a variable reference with an element.

Restricts: oval-def:EntityStateSimpleBaseType

Attributes

Table 98: Attributes

Attribute	Type	Desc.
datatype	oval:SimpleDatatypeEnumeration (required *fixed*='binary')	(No Description)

Simple Content: Union of xsd:hexBinary, oval:EmptyStringType

== EntityStateBoolType ==

The EntityStateBoolType type is extended by the entities of an individual OVAL State. This type provides uniformity to each state entity by including the attributes found in the EntityStateSimpleBaseType. This specific type describes simple boolean data. The empty string is also allowed when using a variable reference with an element.

Restricts: oval-def:EntityStateSimpleBaseType

Attributes

Table 99: Attributes

Attribute	Type	Desc.
datatype	oval:SimpleDatatypeEnumeration (required *fixed*='boolean')	(No Description)

Simple Content: Union of xsd:boolean, oval:EmptyStringType

== EntityStateFloatType ==

The EntityStateFloatType type is extended by the entities of an individual OVAL State. This type provides uniformity to each state entity by including the attributes found in the EntityStateSimpleBaseType. This specific type describes simple float data. The empty string is also allowed when using a variable reference with an element.

Restricts: oval-def:EntityStateSimpleBaseType

Attributes

Table 100: Attributes

Attribute	Type	Desc.
datatype	oval:SimpleDatatypeEnumeration (required *fixed*='float')	(No Description)

Simple Content: Union of xsd:float, oval:EmptyStringType

== EntityStateIntType ==

The EntityStateIntType type is extended by the entities of an individual OVAL State. This type provides uniformity to each state entity by including the attributes found in the EntityStateSimpleBaseType. This specific type describes simple integer data. The empty string is also allowed when using a variable reference with an element.

Restricts: oval-def:EntityStateSimpleBaseType

Attributes

Table 101: Attributes

Attribute	Type	Desc.
datatype	oval:SimpleDatatypeEnumeration (required *fixed*='int')	(No Description)

Simple Content: Union of xsd:integer, oval:EmptyStringType

== EntityStateEVRStringType ==

The EntityStateEVRStringType type is extended by the entities of an individual OVAL State. This type provides uniformity to each state entity by including the attributes found in the EntityStateSimpleBaseType. This type represents the epoch, version, and release fields, for an RPM package, as a single version string. It has the form "EPOCH:VERSION-RELEASE". Note that a null epoch (or '(none)' as returned by rpm) is equivalent to '0' and would hence have the form 0:VERSION-RELEASE. Comparisons involving this datatype should follow the algorithm of librpm's rpmvercmp() function.

Restricts: oval-def:EntityStateSimpleBaseType

Attributes

Table 102: Attributes

Attribute	Type	Desc.
datatype	oval:SimpleDatatypeEnumeration (required *fixed*='evr_string')	(No Description)

Simple Content: Restricts xsd:string

== EntityStateDebianEVRStringType ==

The EntityStateDebianEVRStringType type is extended by the entities of an individual OVAL State. This type provides uniformity to each state entity by including the attributes found in the EntityStateSimpleBaseType. This type represents the epoch, upstream_version, and debian_revision fields, for a Debian package, as a single version string. It has the form "EPOCH:UPSTREAM_VERSION-DEBIAN_REVISION". Note that a null epoch (or '(none)' as returned by dpkg) is equivalent to '0' and would hence have the form 0:UPSTREAM_VERSION-DEBIAN_REVISION. Comparisons involving this datatype should follow the algorithm outlined in Chapter 5 of the "Debian Policy Manual" (https://www.debian.org/doc/debian-policy/ch-controlfields.html#s-f-Version). An implementation of this is the cmpversions() function in dpkg's enquiry.c.

Restricts: oval-def:EntityStateSimpleBaseType

Attributes

Table 103: Attributes

Attribute	Type	Desc.
datatype	oval:SimpleDatatypeEnumeration (required *fixed*='debian_evr_string')	(No Description)

Simple Content: Restricts xsd:string

== EntityStateVersionType ==

The EntityStateVersionType type is extended by the entities of an individual OVAL State. This type provides uniformity to each state entity by including the attributes found in the EntityStateSimpleBaseType. This specific type describes simple version data.

Restricts: oval-def:EntityStateSimpleBaseType

Attributes

Table 104: Attributes

Attribute	Type	Desc.
datatype	oval:SimpleDatatypeEnumeration (required *fixed*='version')	(No Description)

Simple Content: Restricts xsd:string

== EntityStateFileSetRevisionType ==

The EntityStateFileSetRevisionType type is extended by the entities of an individual OVAL State. This type provides uniformity to each state entity by including the attributes found in the EntityStateSimpleBaseType. This specific type represents the version string related to filesets in HP-UX.

Restricts: oval-def:EntityStateSimpleBaseType

Attributes

Table 105: Attributes

Attribute	Type	Desc.
datatype	oval:SimpleDatatypeEnumeration (required *fixed*='fileset_revision')	(No Description)

Simple Content: Restricts xsd:string

== EntityStateIOSVersionType ==

The EntityStateIOSVersionType type is extended by the entities of an individual OVAL State. This type provides uniformity to each state entity by including the attributes found in the EntityStateSimpleBaseType. This specific type represents the version string related to CISCO IOS.

Restricts: oval-def:EntityStateSimpleBaseType

Attributes

Table 106: Attributes

Attribute	Type	Desc.
datatype	Restriction of oval:SimpleDatatypeEnumeration (optional *default*='string') ('ios_version', 'string')	(No Description)

Simple Content: Restricts xsd:string

== EntityStateStringType ==

The EntityStateStringType type is extended by the entities of an individual OVAL State. This type provides uniformity to each state entity by including the attributes found in the EntityStateSimpleBaseType. This specific type describes simple string data.

Restricts: oval-def:EntityStateSimpleBaseType

Attributes

Table 107: Attributes

Attribute	Type	Desc.
datatype	oval:SimpleDatatypeEnumeration (optional *fixed*='string')	(No Description)

Simple Content: Restricts xsd:string

== EntityStateRecordType ==

The EntityStateRecordType defines an entity that consists of a number of uniquely named fields. This structure is used for representing a record from a database query and other similar structures where multiple related fields must be collected at once. Note that for all entities of this type, the only allowed datatype is 'record' and the only allowed operation is 'equals'. During analysis of a system characteristics item, each field is analyzed and then the overall result for elements of this type is computed by logically anding the results for each field and then applying the entity_check attribute.

Note the datatype attribute must be set to 'record'.

Note the operation attribute must be set to 'equals'.

Note the var_ref attribute is not permitted and the var_check attribute does not apply.

Note that when the mask attribute is set to 'true', all child field elements must be masked regardless of the child field's mask attribute value.

Extends: oval-def:EntityStateComplexBaseType

Child Elements

Table 108: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
field	oval-def:EntityStateFieldType (0..unbounded)	

== EntityStateFieldType ==

The EntityStateFieldType defines an element with simple content that represents a named field in a record that may contain any number of named fields. The EntityStateFieldType is much like all other entities with one significant difference, the EntityStateFieldType has a name attribute

The required name attribute specifies a unique name for the field. Field names are lowercase and must be unique within a given parent record element. When analyzing system characteristics an error should be reported for the result of a field that is present in the OVAL State, but not found in the system characteristics Item.

The optional entity_check attribute specifies how to handle multiple record fields with the same name in the OVAL Systems Characteristics file. For example, while collecting group information where one field is the represents the users that are members of the group. It is very likely that there will be multiple fields with a name of 'user' associated with the group. If the OVAL State defines the value of the field with name equal 'user' to equal 'Fred', then the entity_check attribute determines if all values for field entities must be equal to 'Fred', or at least one value must be equal to 'Fred', etc.

Note that when the mask attribute is set to 'true' on a field's parent element the field must be masked regardless of the field's mask attribute value.

Attributes

Table 109: Attributes

Attribute	Type	Desc.
name	Restriction of xsd:string (required)	A string restricted to disallow upper case characters.
entity_check	oval:CheckEnumeration (optional *default*='all')	(No Description)

Simple Content: xsd:anySimpleType

Open Vulnerability and Assessment Language: Core System Characteristics

- Schema: Core System Characteristics
- Version: 5.11.2
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the core schema for encoding Open Vulnerability and Assessment Language (OVAL) System Characteristics. The Core System Characteristics Schema defines all operating system independent objects. These objects are extended and enhanced by individual family schemas, which are described in separate documents. Each of the elements, types, and attributes that make up the Core System Characteristics 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 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.cisecurity.org.

< oval_system_characteristics >

The system_characteristics element is the root of an OVAL System Characteristics Document, and must occur exactly once. Its purpose is to bind together the four major sections of a system characteristics file - generator, system_info, collected_objects, and system_data - which are the children of the oval_system_characteristics element.

Child Elements

Table 110: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
generator	oval:GeneratorType (1..1)	The generator section must be present and provides information about when the system characteristics file was compiled and under what version.
system_info	oval-sc:SystemInfoType (1..1)	The required system_info element is used to record information about the system being described.
collected_objects	oval-sc:CollectedObjectsType (0..1)	The optional collected_objects section is used to associated the ids of the OVAL Objects collected with the system characteristics items that have been defined. The collected_objects section provides a listing of all the objects used to generate this system characteristics file.
system_data	oval-sc:SystemDataType (0..1)	The optional system_data section defines the specific characteristics that have been collected from the system.
ds:Signature	n/a (0..1)	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. More information about the official W3C Recommendation regarding XML digital signatures can be found at http://www.w3.org/TR/xmldsig-core/.

== SystemInfoType ==

The SystemInfoType complex type specifies general information about the system that data was collected from, including information that can be used to identify the system. See the description of the InterfacesType complex type for more information. Note that the high level interfaces is required due to the inclusion of the xsd:any tag that follows it. The interfaces tag can be empty if no single interface is present.

Additional system information is also allowed although it is not part of the official OVAL Schema. Individual organizations can place system information that they feel is important and these will be skipped during the validation. All OVAL really cares about is that the required system information items are there.

Child Elements

Table 111: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
os_name	xsd:string (1..1)	The required os_name elements describes the operating system of the machine the data was collected on.
os_version	xsd:string (1..1)	The required os_version elements describe the operating system version of the machine the data was collected on.
architecture	xsd:string (1..1)	The required architecture element describes the hardware architecture type of the system data was collected on.
primary_host_name	xsd:string (1..1)	The required primary_host_name element is the primary host name of the machine the data was collected on.
interfaces	oval-sc:InterfacesType (1..1)	The required interfaces element outlines the network interfaces that exist on the system.
xsd:any	n/a (0..unbounded)	The Asset Identification specification (http://scap.nist.gov/specifications/ai/) provides a standardized way of reporting asset information across different organizations.The information contained within an AI computing-device element is similar to the information collected by OVAL's SystemInfoType.To support greater interoperability, an ai:computing-device element describing the system that data was collected from may appear at this point in an OVAL System Characteristics document.

== InterfacesType ==

The InterfacesType complex type is a container for zero or more interface elements. Each interface element is used to describe an existing network interface on the system.

Child Elements

Table 112: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
interface	oval-sc:InterfaceType (0..unbounded)	Please refer to the description of the InterfaceType for more information.

== InterfaceType ==

The InterfaceType complex type is used to describe an existing network interface on the system. This information can help identify a specific system on a given network.

Child Elements

Table 113: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
interface_name	xsd:string (1..1)	The required interface_name element is the name of the interface
ip_address	xsd:string (1..1)	The required ip_address element holds the IP address for the interface. Note that the IP address can be IPv4 or IPv6.
mac_address	xsd:string (1..1)	The required mac_address element holds the MAC address for the interface. MAC addresses should be formatted according to the IEEE 802-2001 standard which states that a MAC address is a sequence of six octet values, separated by hyphens, where each octet is represented by two hexadecimal digits. Uppercase letters should also be used to represent the hexadecimal digits A through F.

== CollectedObjectsType ==

The CollectedObjectsType complex type states all the objects that have been collected by the system characteristics file. The details of each object are defined by the global OVAL object that is identified by the id.

Child Elements

Table 114: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-sc:ObjectType (1..unbounded)	

== ObjectType ==

The ObjectType complex type provides a reference between items collected and a related global OVAL Object.

If an OVAL Object does not exist on the system, then an object element is still provided but with the flag attribute set to 'does not exist'. For details on how to handle items, when an OVAL Object does not exist on the system, please see the ItemType documentation. This shows that the object was looked for but not found on the system. If no object element is written in this case, users of the system characteristics file will not know whether the object was not found or no attempt was made to collect it.

The required id attribute is the id of the global OVAL Object.

The required version attribute is the specific version of the global OVAL Object that was used by the data collection engine. The version is necessary so that analysis using a system characteristics file knows exactly what was collected.

The optional variable_instance identifier is a unique id that differentiates each unique instance of an object. Capabilities that use OVAL may reference the same definition multiple times and provide different variable values each time the definition is referenced. This will result in multiple instances of an object being included in the OVAL System Characteristics file (definitions that do not use variables can only have one unique instance). The inclusion of this unique instance identifier allows the OVAL Results document to associate the correct objects and items for each combination of supplied values.

The optional comment attribute provides a short description of the object.

The required flag attribute holds information regarding the outcome of the data collection. For example, if there was an error looking for items that match the object specification, then the flag would be ‘error’. Please refer to the description of FlagEnumeration for details about the different flag values.

Attributes

Table 115: Attributes

Attribute	Type	Desc.
id	oval:ObjectIDPattern (required)	(No Description)
version	xsd:nonNegativeInteger (required)	(No Description)
variable_instance	xsd:nonNegativeInteger (optional *default*='1')	(No Description)
comment	xsd:string (optional)	(No Description)
flag	oval-sc:FlagEnumeration (required)	(No Description)

Child Elements

Table 116: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
message	oval:MessageType (0..unbounded)	The optional message element holds an error message or some other string that the data collection engine wishes to pass along.
variable_value	oval-sc:VariableValueType (0..unbounded)	The optional variable_value elements define the actual value(s) used during data collection of any variable referenced by the object (as well as any object referenced via a set element). An OVAL Object that includes a variable maybe have a different unique set of matching items depending on the value assigned to the variable. A tool that is given an OVAL System Characteristics file in order to analyze an OVAL Definition needs to be able to determine the exact instance of an object to use based on the variable values supplied. If a variable represents a collection of values, then multiple variable_value elements would exist with the same variable_id attribute.
reference	oval-sc:ReferenceType (0..unbounded)	The optional reference element links the collected item found by the data collection engine and the global OVAL Object. A global OVAL Object my have multiple matching items on a system. For example a global file object that is a pattern match might match 10 different files on a specific system. In this case, there would be 10 reference elements, one for each of the files found on the system.

== VariableValueType ==

The VariableValueType complex type holds the value to a variable used during the collection of an object. The required variable_id attribute is the unique id of the variable being identified.

Attributes

Table 117: Attributes

Attribute	Type	Desc.
variable_id	oval:VariableIDPattern (required)	(No Description)

Simple Content: xsd:anySimpleType

== ReferenceType ==

The ReferenceType complex type specifies an item in the system characteristics file. This reference is used to link global OVAL Objects to specific items.

Attributes

Table 118: Attributes

Attribute	Type	Desc.
item_ref	oval:ItemIDPattern (required)	(No Description)

== SystemDataType ==

The SystemDataType complex type is a container for one or more item elements. Each item defines a specific piece of data on the system.

Child Elements

Table 119: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
oval-sc:item	n/a (1..unbounded)	

< item >

The abstract item element holds information about a specific item on a system. An item might be a file, a rpm, a process, etc. This element is extended by the different component schemas through substitution groups. Each item represents a unique instance of an object as specified by an OVAL Object. For example, a single file or a single user. Each item may be referenced by more than one object in the collected object section. Please refer to the description of ItemType for more details about the information stored in items.

oval-sc:ItemType

== ItemType ==

The ItemType complex type specifies an optional message element that is used to pass things like error messages during data collection to a tool that will utilize the information.

The required id attribute is a unique (to the file) identifier that allows the specific item to be referenced.

The required status attribute holds information regarding the success of the data collection. For example, if an item exists on the system then the status would reflect this with a value of 'exists'. If an error occurs which is not associated with any item entities, or if an error occurs that is associated with an item entity matching an associated object entity, then the status would be 'error'. An error specific to any particular entity should be addressed at the entity level and, for item entities not associated with an object entity, not the item level. When creating items, any entities that can successfully be collected should be reported.

In some cases, when an item for a specified object does not exist, it may be beneficial to report a partial match of an item showing what entities did exist and what entities did not exist for debugging purposes. This is especially true when considering items that are collected by objects with hierarchical object entities. An example of such a case is when a file_object has a path entity equal to 'C:' and a filename entity equal to 'test.txt' where 'test.txt' does not exist in the 'C:' directory. This would result in the creation of a partially matching file_item with a status of 'does not exist' where the path entity equals 'C:' and the filename entity equals 'test.txt' with a status of 'does not exist'. By showing the partial match, someone reading a system-characteristics document can quickly see that a matching file_item did not exist because the specified filename did not exist and not that the specified path did not exist. Again, please note that the implementation of partial matches, when an item for a specified object does not exist, is completely optional.

Attributes

Table 120: Attributes

Attribute	Type	Desc.
id	oval:ItemIDPattern (required)	(No Description)
status	oval-sc:StatusEnumeration (optional *default*='exists')	(No Description)

Child Elements

Table 121: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
message	oval:MessageType (0..50)	

– FlagEnumeration –

The FlagEnumeration simple type defines the valid flags associated with a collected object. These flags are meant to provide information about how the specified object was handled by the data collector. In order to evaluate an OVAL Definition, information about the defined objects needs to be available. The flags help detail the outcome of attempting to collect information related to these objects..

Table 122: Enumeration Values

Value	Description
error	A flag of 'error' indicates that there was an error trying to identify items on the system that match the specified object declaration. This flag is not meant to be used when there was an error retrieving a specific entity, but rather when it could not be determined if an item exists or not. Any error in retrieving a specific entity should be represented by setting the status of that specific entity to 'error'.
complete	A flag of 'complete' indicates that every matching item on the system has been identified and is represented in the system characteristics file. It can be assumed that no additional matching items exist on the system.
incomplete	A flag of 'incomplete' indicates that a matching item exists on the system, but only some of the matching items have been identified and are represented in the system characteristics file. It is unknown if additional matching items also exist. Note that with a flag of 'incomplete', each item that has been identified matches the object declaration, but additional items might also exist on the system.
does not exist	A flag of 'does not exist' indicates that the underlying structure is installed on the system but no matching item was found. For example, the Windows metabase is installed but there were no items that matched the metabase_object. In this example, if the metabase itself was not installed, then the flag would have been 'not applicable'.
not collected	A flag of 'not collected' indicates that no attempt was made to collect items on the system. An object with this flag will produce an 'unknown' result during analysis since it is unknown if matching items exists on the system or not. This is different from an 'error' flag because an 'error' flag indicates that an attempt was made to collect items on system whereas a 'not collected' flag indicates that an attempt was not made to collect items on the system.
not applicable	A flag of 'not applicable' indicates that the specified object is not applicable to the system being characterized. This could be because the data repository is not installed or that the object structure is for a different flavor of systems. An example would be trying to collect objects related to a Red Hat system off of a Windows system. Another example would be

Below is a table that outlines how each FlagEnumeration value effects evaluation of a given test. Note that this is related to the existence of a unique set of items identified by an object and not each item’s compliance with a state. The left column identifies the FlagEnumeration value in question. The right column specifies the ResultEnumeration value that should be used when evaluating the collected object. ‘‘‘

||

flag value || test result is ||

——————||——————————- error || error complete || (test result depends on incomplete || check_existence and does not exist || check attributes) not collected || unknown not applicable || not applicable

——————||—————————— ‘‘‘

– StatusEnumeration –

The StatusEnumeration simple type defines the valid status messages associated with collection of specific information associated with an item.

Table 123: Enumeration Values

Value	Description
error	A status of ‘error’ says that there was an error collecting information associated with an item as a whole or any specific entity. An item would have a status of ‘error’ if a problem occurred that prevented the item from being collected. For example, a file_item would have a status of ‘error’ if a handle to the file could not be opened because the handle was already in use by another program. See the documentation for ItemType for information about when an item entity status of ‘error’ should propagate up to the item status level.
exists	A status of ‘exists’ says that the item or specific piece of information exists on the system and has been collected.
does not exist	A status of ‘does not exist’ says that the item or specific piece of information does not exist and therefore has not been collected. This status assumes that an attempt was made to collect the information, but the information just does not exist. This can happen when a certain entity is only pertinent to particular instances or if the information for that entity is not set.
not collected	A status of ‘not collected’ says that no attempt was made to collect the item or specific piece of information so it is unknown what the value is and if it even exists.

– EntityAttributeGroup –

The EntityAttributeGroup is a collection of attributes that are common to all entities. This group defines these attributes and their default values. Individual entities may limit allowed values for these attributes, but all entities will support these attributes.

Attributes

Table 124: Attributes

Attribute	Type	Desc.
datatype	oval:DatatypeEnumeration (optional *default*=’string’)	The optional datatype attribute determines the type of data expected (the default datatype is ‘string’). Note that the datatype attribute simply defines the type of data as found on the system, it is not used during evaluation. An OVAL Definition defines how the data should be interpreted during analysis. If the definition states a datatype that is different than what the system characteristics presents, then a type cast must be made.
mask	xsd:boolean (optional *default*=’false’)	The optional mask attribute is used to identify values that have been hidden for sensitivity concerns. This is used by the Result document which uses the System Characteristics schema to format the information found on a specific system. When the mask attribute is set to ‘true’ on an OVAL Entity or an OVAL Field, the corresponding collected value of that OVAL Entity or OVAL Field MUST NOT be present in the “results” section of the OVAL Results document; the “oval_definitions” section must not be altered and must be an exact copy of the definitions evaluated. Values MUST NOT be masked in OVAL System Characteristics documents that are not contained within an OVAL Results document. It is possible for masking conflicts to occur where one entity has mask set to true and another entity has mask set to false. A conflict will occur when the mask attribute is set differently on an OVAL Object and matching OVAL State or when more than one OVAL Objects identify the same OVAL Item(s). When such a conflict occurs the result is always to mask the entity.
status	oval-sc:StatusEnumeration (optional *default*=’exists’)	The optional status attribute holds information regarding the success of the data collection. For Example, if there was an error collecting a particular piece of data, then the status would be ‘error’.

== EntityItemSimpleBaseType ==

The EntityItemSimpleBaseType complex type is an abstract type that serves as the base type for all simple item entities.

Simple Content: xsd:anySimpleType

== EntityItemComplexBaseType ==

The EntityItemComplexBaseType complex type is an abstract type that serves as the base type for all complex item entities.

== EntityItemIPAddressType ==

The EntityItemIPAddressType type is extended by the entities of an individual item. This type provides uniformity to each entity by including the attributes found in the EntityItemSimpleBaseType. This specific type describes any IPv4/IPv6 address or address prefix.

Restricts: oval-sc:EntityItemSimpleBaseType

Attributes

Table 125: Attributes

Attribute	Type	Desc.
datatype	Restriction of oval:SimpleDatatypeEnumeration (required) (‘ipv4_address’, ‘ipv6_address’)	(No Description)

Simple Content: Restricts xsd:string

== EntityItemIPAddressStringType ==

The EntityItemIPAddressStringType type is extended by the entities of an individual item. This type provides uniformity to each entity by including the attributes found in the EntityItemSimpleBaseType. This specific type describes any IPv4/IPv6 address, address prefix, or its string representation.

Restricts: oval-sc:EntityItemSimpleBaseType

Attributes

Table 126: Attributes

Attribute	Type	Desc.
datatype	Restriction of oval:SimpleDatatypeEnumeration (optional *default*=’string’) (‘ipv4_address’, ‘ipv6_address’, ‘string’)	(No Description)

Simple Content: Restricts xsd:string

== EntityItemAnySimpleType ==

The EntityItemAnySimpleType type is extended by the entities of an individual item. This type provides uniformity to each entity by including the attributes found in the EntityItemSimpleBaseType. This specific type describes any simple data.

Restricts: oval-sc:EntityItemSimpleBaseType

Attributes

Table 127: Attributes

Attribute	Type	Desc.
datatype	oval:SimpleDatatypeEnumeration (optional *default*=’string’)	(No Description)

Simple Content: Restricts xsd:string

== EntityItemBinaryType ==

The EntityItemBinaryType type is extended by the entities of an individual item. This type provides uniformity to each entity by including the attributes found in the EntityItemSimpleBaseType. This specific type describes simple binary data. The empty string is also allowed for cases where there was an error in the data collection of an entity and a status needs to be reported.

Restricts: oval-sc:EntityItemSimpleBaseType

Attributes

Table 128: Attributes

Attribute	Type	Desc.
datatype	oval:SimpleDatatypeEnumeration (required *fixed*='binary')	(No Description)

Simple Content: Union of xsd:hexBinary, oval:EmptyStringType

== EntityItemBoolType ==

The EntityItemBoolType type is extended by the entities of an individual item. This type provides uniformity to each entity by including the attributes found in the EntityItemSimpleBaseType. This specific type describes simple boolean data. The empty string is also allowed for cases where there was an error in the data collection of an entity and a status needs to be reported.

Restricts: oval-sc:EntityItemSimpleBaseType

Attributes

Table 129: Attributes

Attribute	Type	Desc.
datatype	oval:SimpleDatatypeEnumeration (required *fixed*='boolean')	(No Description)

Simple Content: Union of xsd:boolean, oval:EmptyStringType

== EntityItemFloatType ==

The EntityItemFloatType type is extended by the entities of an individual item. This type provides uniformity to each entity by including the attributes found in the EntityItemSimpleBaseType. This specific type describes simple float data. The empty string is also allowed for cases where there was an error in the data collection of an entity and a status needs to be reported.

Restricts: oval-sc:EntityItemSimpleBaseType

Attributes

Table 130: Attributes

Attribute	Type	Desc.
datatype	oval:SimpleDatatypeEnumeration (required *fixed*='float')	(No Description)

Simple Content: Union of xsd:float, oval:EmptyStringType

== EntityItemIntType ==

The EntityItemIntType type is extended by the entities of an individual item. This type provides uniformity to each entity by including the attributes found in the EntityItemSimpleBaseType. This specific type describes simple integer data. The empty string is also allowed for cases where there was an error in the data collection of an entity and a status needs to be reported.

Restricts: oval-sc:EntityItemSimpleBaseType

Attributes

Table 131: Attributes

Attribute	Type	Desc.
datatype	oval:SimpleDatatypeEnumeration (required *fixed*='int')	(No Description)

Simple Content: Union of xsd:integer, oval:EmptyStringType

== EntityItemStringType ==

The EntityItemStringType type is extended by the entities of an individual item. This type provides uniformity to each entity by including the attributes found in the EntityItemSimpleBaseType. This specific type describes simple string data.

Restricts: oval-sc:EntityItemSimpleBaseType

Attributes

Table 132: Attributes

Attribute	Type	Desc.
datatype	oval:SimpleDatatypeEnumeration (optional *fixed*='string')	(No Description)

Simple Content: Restricts xsd:string

== EntityItemRecordType ==

The EntityItemRecordType defines an entity that consists of a number of named fields. This structure is used for representing a record from a database query and other similar structures where multiple related fields must be collected at once. Note that for all entities of this type, the only allowed datatype is 'record'.

Note the datatype attribute must be set to ‘record’.

Note that when the mask attribute is set to ‘true’, all child field elements must be masked regardless of the child field’s mask attribute value.

Extends: oval-sc:EntityItemComplexBaseType

Child Elements

Table 133: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
field	oval-sc:EntityItemFieldType (0..unbounded)	

== EntityItemFieldType ==

The EntityItemFieldType defines an element with simple content that represents a named field in a record that may contain any number of named fields. The EntityItemFieldType is much like all other entities with one significant difference, the EntityItemFieldType has a name attribute.

The required name attribute specifies a name for the field. Field names are lowercase and may occur more than once to allow for a field to have multiple values.

Note that when the mask attribute is set to ‘true’ on a field’s parent element the field must be masked regardless of the field’s mask attribute value.

Attributes

Table 134: Attributes

Attribute	Type	Desc.
name	Restriction of xsd:string (required)	A string restricted to disallow upper case characters.

Simple Content: xsd:anySimpleType

== EntityItemVersionType ==

The EntityItemVersionType type is extended by the entities of an individual item. This type provides uniformity to each entity by including the attributes found in the EntityItemSimpleBaseType. This specific type describes version data.

Restricts: oval-sc:EntityItemSimpleBaseType

Attributes

Table 135: Attributes

Attribute	Type	Desc.
datatype	oval:SimpleDatatypeEnumeration (required *fixed*=’version’)	(No Description)

Simple Content: Restricts xsd:string

== EntityItemFilesetRevisionType ==

The EntityItemFilesetRevisionType type is extended by the entities of an individual item. This type provides uniformity to each entity by including the attributes found in the EntityItemSimpleBaseType. This specific type represents the version string related to filesets in HP-UX.

Restricts: oval-sc:EntityItemSimpleBaseType

Attributes

Table 136: Attributes

Attribute	Type	Desc.
datatype	oval:SimpleDatatypeEnumeration (required *fixed*='fileset_revision')	(No Description)

Simple Content: Restricts xsd:string

== EntityItemIOSVersionType ==

The EntityItemIOSVersionType type is extended by the entities of an individual item. This type provides uniformity to each entity by including the attributes found in the EntityItemSimpleBaseType. This specific type represents the version string for IOS.

Restricts: oval-sc:EntityItemSimpleBaseType

Attributes

Table 137: Attributes

Attribute	Type	Desc.
datatype	oval:SimpleDatatypeEnumeration (required *fixed*='ios_version')	(No Description)

Simple Content: Restricts xsd:string

== EntityItemEVRStringType ==

The EntityItemEVRStringType type is extended by the entities of an individual item. This type provides uniformity to each entity by including the attributes found in the EntityItemSimpleBaseType. This type represents the epoch, version, and release fields, for an RPM package, as a single version string. It has the form "EPOCH:VERSION-RELEASE". Note that a null epoch (or '(none)' as returned by rpm) is equivalent to '0' and would hence have the form 0:VERSION-RELEASE. Comparisons involving this datatype should follow the algorithm of librpm's rpmvercmp() function.

Restricts: oval-sc:EntityItemSimpleBaseType

Attributes

Table 138: Attributes

Attribute	Type	Desc.
datatype	oval:SimpleDatatypeEnumeration (required *fixed*='evr_string')	(No Description)

Simple Content: Restricts xsd:string

== EntityItemDebianEVRStringType ==

The EntityItemDebianEVRStringType type is extended by the entities of an individual item. This type provides uniformity to each entity by including the attributes found in the EntityItemSimpleBaseType. This type represents the epoch, upstream_version, and debian_revision fields, for a Debian package, as a single version string. It has the form "EPOCH:UPSTREAM_VERSION-DEBIAN_REVISION". Note that a null epoch (or '(none)' as returned by dpkg) is equivalent to '0' and would hence have the form 0:UPSTREAM_VERSION-DEBIAN_REVISION. Comparisons involving this datatype should follow the algorithm outlined in Chapter 5 of the "Debian Policy Manual" (https://www.debian.org/doc/debian-policy/ch-controlfields.html#s-f-Version). An implementation of this is the cmpversions() function in dpkg's enquiry.c.

Restricts: oval-sc:EntityItemSimpleBaseType

Attributes

Table 139: Attributes

Attribute	Type	Desc.
datatype	oval:SimpleDatatypeEnumeration (required *fixed*='debian_evr_string')	(No Description)

Simple Content: Restricts xsd:string

Open Vulnerability and Assessment Language: Core Results

- Schema: Core Results
- Version: 5.11.2
- Release Date: 11/30/2016 09:00:00 AM

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 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.cisecurity.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 document - 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.

Child Elements

Table 140: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
generator	oval:GeneratorType (1..1)	The required generator section provides information about when the results document was compiled and under what version.
directives	oval-res:DefaultDirectivesType (1..1)	The required directives section presents flags describing what information has been included in the results document. This element represents the default set of directives. These directives apply to all classes of definitions for which there is not a class specific set of directives.
class_directives	oval-res:ClassDirectivesType (0..5)	The optional class_directives section presents flags describing what information has been included in the results document for a specific OVAL Definition class. The directives for a particular class override the default directives. Using OVAL Results class_directives, an OVAL Results document dealing with vulnerabilities might by default include only minimal information and then include full details for all vulnerability definitions that evaluated to true.
oval-def:oval_definitions	n/a (0..1)	The oval_definitions section is optional and dependent on the include_source_definitions attribute of the directives element. Its purpose is to provide an exact copy of the definitions evaluated for the results document.
results	oval-res:ResultsType (1..1)	The required results section holds all the results of the evaluated definitions.
ds:Signature	n/a (0..1)	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. More information about the official W3C Recommendation regarding XML digital signatures can be found at http://www.w3.org/TR/xmldsig-core/.

== DirectivesType ==

The DirectivesType complex type presents a set of flags that describe what information has been included in the results document. There are six possible results (true, false, unknown, error, not evaluated, and not applicable) for the evaluation of an OVAL Definition. The directives state which of these results are being reported in the results document.

Child Elements

Table 141: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
definition_true	oval-res:DirectiveType (1..1)	
definition_false	oval-res:DirectiveType (1..1)	
definition_unknown	oval-res:DirectiveType (1..1)	
definition_error	oval-res:DirectiveType (1..1)	
definition_not_evaluated	oval-res:DirectiveType (1..1)	
definition_not_applicable	oval-res:DirectiveType (1..1)	

== DefaultDirectivesType ==

The DefaultDirectivesType complex type presents the default set of flags that describe what information has been included in the results document. See the definition of the oval-res:DirectivesType for more information.

The optional include_source_definitions attribute indicates whether or not the source OVAL Definitions document has been included in the results document. A value of false indicates that the source OVAL Definitions has not been included. By default the source document is included.

Extends: oval-res:DirectivesType

Attributes

Table 142: Attributes

Attribute	Type	Desc.
include_source_definitions	xsd:boolean (optional *default*='true')	(No Description)

== ClassDirectivesType ==

The ClassDirectivesType complex type presents a set of flags that describe what information has been included in the results document for a specific OVAL Definition class. See the definition of the oval-res:DirectivesType for more information.

The required class attribute allows a set of directives to be specified for each supported OVAL Definition class (See the definition of the oval:ClassEnumeration for more information about the supported classes). A set of class specific directives overrides the default directives for the specified definition class. A given class may be specified once.

Extends: oval-res:DirectivesType

Attributes

Table 143: Attributes

Attribute	Type	Desc.
class	oval:ClassEnumeration (required)	(No Description)

== DirectiveType ==

An individual directive element determines whether or not a specific type of result is included in the results document. The required reported 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 results for all the individual tests and extended definitions. Please refer to the oval-res:ContentEnumeration for details about the different content options.

Attributes

Table 144: Attributes

Attribute	Type	Desc.
reported	xsd:boolean (required)	(No Description)
content	oval-res:ContentEnumeration (optional *default*='full')	(No Description)

== 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

Table 145: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
system	oval-res:SystemType (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 document used to perform the evaluation of the OVAL Definitions.

Child Elements

Table 146: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
definitions	oval-res:DefinitionsType (0..1)	
tests	oval-res:TestsType (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

Table 147: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
definition	oval-res:DefinitionType (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 each unique instance of a definition. Capabilities that use OVAL may reference the same definition multiple times and provide different variable values each time the definition is referenced. This will result in multiple instances of a definition being included in the OVAL Results document (definitions that do not use variables can only have one unique instance). The inclusion of this unique instance identifier allows the OVAL Results document to associate the correct objects and items for each combination of supplied values.

The optional class attribute …

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

Table 148: Attributes

Attribute	Type	Desc.
definition_id	oval:DefinitionIDPattern (required)	(No Description)
version	xsd:nonNegativeInteger (required)	(No Description)
variable_instance	xsd:nonNegativeInteger (optional *default*='1')	(No Description)
class	oval:ClassEnumeration (optional)	(No Description)
result	oval-res:ResultEnumeration (required)	(No Description)

Child Elements

Table 149: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
message	oval:MessageType (0..unbounded)	
criteria	oval-res:CriteriaType (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 and 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.

The optional applicability_check attribute provides a Boolean flag that when true indicates that the criteria is being used to determine whether the OVAL Definition applies to a given system.

Attributes

Table 150: Attributes

Attribute	Type	Desc.
applicability_check	xsd:boolean (optional)	(No Description)
operator	oval:OperatorEnumeration (required)	(No Description)
negate	xsd:boolean (optional *default*='false')	(No Description)
result	oval-res:ResultEnumeration (required)	(No Description)

Child Elements

Table 151: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
criteria	oval-res:CriteriaType (1..unbounded)	
criterion	oval-res:CriterionType (1..unbounded)	
extend_definition	oval-res:ExtendDefinitionType (1..unbounded)	

== CriterionType ==

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

The optional applicability_check attribute provides a Boolean flag that when true indicates that the criterion is being used to determine whether the OVAL Definition applies to a given system.

The required test_ref 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 variable values 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

Table 152: Attributes

Attribute	Type	Desc.
applicability_check	xsd:boolean (optional)	(No Description)
test_ref	oval:TestIDPattern (required)	(No Description)
version	xsd:nonNegativeInteger (required)	(No Description)
variable_instance	xsd:nonNegativeInteger (optional *default*='1')	(No Description)
negate	xsd:boolean (optional *default*='false')	(No Description)
result	oval-res:ResultEnumeration (required)	(No Description)

== ExtendDefinitionType ==

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

The optional applicability_check attribute provides a Boolean flag that when true indicates that the extend_definition is being used to determine whether the OVAL Definition applies to a given system.

The required definition_ref 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 attribute is a unique id that differentiates each unique instance of a definition. Capabilities that use OVAL may reference the same definition multiple times and provide different variable values each time the definition is referenced. This will result in multiple instances of a definition being included in the OVAL Results document (definitions that do not use variables can only have one unique instance). The inclusion of this unique instance identifier allows the OVAL Results document to associate the correct objects and items 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 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

Table 153: Attributes

Attribute	Type	Desc.
applicability_check	xsd:boolean (optional)	(No Description)
definition_ref	oval:DefinitionIDPattern (required)	(No Description)
version	xsd:nonNegativeInteger (required)	(No Description)
variable_instance	xsd:nonNegativeInteger (optional *default*='1')	(No Description)
negate	xsd:boolean (optional *default*='false')	(No Description)
result	oval-res:ResultEnumeration (required)	(No Description)

== 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

Table 154: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
test	oval-res:TestType (1..unbounded)	

== TestType ==

The TestType complex type provides a reference to every item that matched the object section of the original test as well as providing an overall test result based on those items. The optional message element holds an error message or some other string that the analysis engine wishes to pass along. 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 a collection of values, then multiple tested_variable elements would exist with the same variable_id attribute. Please refer to the description of oval-res:TestedVariableType for more information.

The required test_id attribute identifies the test and must conform to the format specified by the oval: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 check_existence, check, and state_operator attributes reflect the values that were specified on the test as it was evaluated. These evaluation control attributes are copied into the OVAL Results file to enable post processing of results documents. More information on each of these attributes is provided with the definition of the oval-def:TestType.

The required result attribute holds the result of the evaluation after all referenced items have been examined and the evaluation control attributes have been applied. Please refer to the description of the oval-res:ResultEnumeration for details about the different result values. In general, the overall result of an OVAL Test is determined by combining the results of each matching item based first on the check_existence attribute, then the check attribute, and finally the state_operator attribute.

The following section provides a more detailed description of how the result for an OVAL Test is determined when using an OVAL System Characteristics document. An OVAL System Characteristics document can contain an optional collected_objects section. When the collected_objects section is present the following rules specify how the overall result for an OVAL Test is determined: When an oval-sc:collected_objects/oval-sc:object with an id that matches the OVAL Object id that is referenced by the OVAL Test is not found, the result for the OVAL Test must be “unknown”. When the flag attribute of the corresponding oval-sc:collected_objects/oval-sc:object is “error”, the result of the OVAL Test must be “error”. When the flag attribute of the corresponding oval-sc:collected_objects/oval-sc:object is “not collected”, the result of the OVAL Test must be “unknown”. When the flag attribute of the corresponding oval-sc:collected_objects/oval-sc:object is “not applicable”, the result of the OVAL Test must be “not applicable”. When the flag attribute of the corresponding oval-sc:collected_objects/oval-sc:object is “does not exist”, the result of the OVAL Test is determined by examining the check_existence attribute’s value and if the check_existence attribute is “none_exist” or “any_exist” the OVAL Test should evaluate to “true”, for all other values of the check_existence attribute the OVAL Test should evaluate to “false”. The check and state_operator attributes do not need to be considered in this condition. When the flag attribute of the corresponding oval-sc:collected_objects/oval-sc:object is “complete”, the result of the OVAL Test is determined by first evaluating the check_existence attribute specified by the OVAL Test and then evaluating the check and state_operator attributes. The check attribute only needs to be considered if the result of evaluating the check_existence attribute is “true”. When the flag attribute of the corresponding oval-sc:collected_objects/oval-sc:object is “incomplete”, the result of the OVAL Test must be “unknown” with the following exceptions: 1) When the check_existence attribute of the OVAL Test is set to “none_exist” and the collected object has 1 or more item references with a status of “exists”, a result of “false” must be reported; 2) When the check_existence attribute of the OVAL Test is set to “only_one_exists”, the collected object has more than 1 item reference with a status of “exists”, a result of “false” must be reported; 3) If after evaluating the check_existence attribute a non “true” result has not been determined, the check attribute must be considered as follows: 3a) If the check attribute evaluation results in “false”, then the OVAL Test result must be “false”; 3b) If the check attribute is set to “at_least_one_satisfies” and its evaluation results in “true”, the OVAL Test result must be “true”. When the collected_objects section is not present in the OVAL System Characteristics document, the evaluation engine must search the system characteristics for all Items that match the OVAL Object referenced by the OVAL Test. The set of matching OVAL Items is then evaluated first based on the check_existence attribute, then the check attribute, and finally the state_operator attribute.

Attributes

Table 155: Attributes

Attribute	Type	Desc.
test_id	oval:TestIDPattern (required)	(No Description)
version	xsd:nonNegativeInteger (required)	(No Description)
variable_instance	xsd:nonNegativeInteger (optional *default*=’1’)	(No Description)
check_existence	oval:ExistenceEnumeration (optional *default*=’at_least_one_exists’)	(No Description)
check	oval:CheckEnumeration (required)	(No Description)
state_operator	oval:OperatorEnumeration (optional *default*=’AND’)	(No Description)
result	oval-res:ResultEnumeration (required)	(No Description)

Child Elements

Table 156: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
message	oval:MessageType (0..unbounded)	
tested_item	oval-res:TestedItemType (0..unbounded)	
tested_variable	oval-res:TestedVariableType (0..unbounded)	

== TestedItemType ==

The TestedItemType complex type holds a reference to a system characteristic item that matched the object specified in a test. Details of the item can be found in the oval_system_characteristics section of the OVAL Results document by using the required item_id. The optional message element holds an error message or some other message 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. If the test did not include a state reference then the result attribute will be set to 'not evaluated'. Please refer to the description of the ResultEnumeration for details about the different result values.

Attributes

Table 157: Attributes

Attribute	Type	Desc.
item_id	oval:ItemIDPattern (required)	(No Description)
result	oval-res:ResultEnumeration (required)	(No Description)

Child Elements

Table 158: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
message	oval:MessageType (0..unbounded)	

== TestedVariableType ==

The TestedVariableType complex type holds the value of 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 documents. If a variable is represented by a collection 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

Table 159: Attributes

Attribute	Type	Desc.
variable_id	oval:VariableIDPattern (required)	(No Description)

Simple Content: xsd:anySimpleType

– ContentEnumeration –

The ContentEnumeration defines the valid values for the directives controlling the amount of expected depth found in the results document. Each directive specified at the top of an OVAL Results document defines how much information should be included in the document for each of the different result types. The amount of content that is expected with each value is defined by Schematron statements embedded throughout the OVAL Results Schema. Currently, the enumeration defines two values: thin and full. Please refer to the documentation of each individual value of this enumeration for more information about what each means.

Table 160: Enumeration Values

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 must be presented.

– ResultEnumeration –

The ResultEnumeration defines the acceptable result values for the DefinitionType, CriteriaType, CriterionType, ExtendDefinitionType, TestType, and TestedItemType constructs.

Table 161: Enumeration Values

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 document. When evaluating a tested_item, and a state exists, a result value of ‘true’ indicates that the item matches the state.
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 document. When evaluating a tested_item, and a state exists, a result value of ‘false’ indicates that the item does not match the state.
unknown	When evaluating a definition or test, a result value of ‘unknown’ means that the characteristics being evaluated cannot be found in the system characteristic document (or the characteristics can be found but collected object flag is ‘not collected’). For example, assume that a definition tests a file, but data pertaining to that file cannot be found and is not recorded in the System Characteristics document. The lack of an item (in the system_data section) for this file in the System Characteristics document means that no attempt was made to collect information about the file. In this situation, there is no way of knowing what the result would be if the file was collected. Note that finding a collected_object element in the system characteristic document is not the same as finding a matching element of the system. When evaluating an OVAL Test, the lack of a matching object on a system (for example, file not found) does not cause a result of unknown since an test considers both the state of an item and its existence. In this case the test result would be based on the existence check specified by the check_existence attribute on the test. When evaluating a tested_item, and a state exists, a result value of ‘unknown’ indicates that it could not be determined whether or not the item and state match. For example, if a registry_object with a hive equal to HKEY_LOCAL_MACHINE, a key with the xsi:nil attribute set to ‘true’, and a name with the xsi:nil attribute set to ‘true’ was collected and compared against a registry_state with key entity equal to ‘SOFTWARE’, the tested_item result would be ‘unknown’ because an assertion of whether or not the item matches the state could not be determined since the key entity of the item was not collected.
error	When evaluating a definition or test, a result value of ‘error’ means that the characteristics being evaluated exist in the system characteristic document but there

Open Vulnerability and Assessment Language: Core Directives

- Schema: Core Directives
- Version: 5.11.2
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the core schema for encoding Open Vulnerability and Assessment Language (OVAL) Directives. Each of the elements, types, and attributes that make up the Core Directives 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_directives >

The oval_directives element is the root of an OVAL Directive Document. Its purpose is to bind together the generator and the set of directives contained in the document. The generator section must be present and provides information about when the directives document was compiled and under what version. 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. More information about the official W3C Recommendation regarding XML digital signatures can be found at http://www.w3.org/TR/xmldsig-core/.

Child Elements

Table 162: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
generator	oval:GeneratorType (1..1)	The required generator section provides information about when the directives document was compiled and under what version.
directives	oval-res:DefaultDirectivesType (1..1)	The required directives section presents flags describing what information must be been included in the oval results document. This element represents the default set of directives. These directives apply to all classes of definitions for which there is not a class specific set of directives.
class_directives	oval-res:ClassDirectivesType (0..5)	The optional class_directives section presents flags describing what information has been included in the results document for a specific OVAL Definition class. The directives for a particlar class override the default directives.
ds:Signature	n/a (0..1)	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. More information about the official W3C Recommendation regarding XML digital signatures can be found at http://www.w3.org/TR/xmldsig-core/.

Open Vulnerability and Assessment Language: OVAL Definition Interpreter - Evaluation Id Schema

- Schema: OVAL Definition Interpreter - Evaluation Id Schema
- Version: 5.11.2
- Release Date: 11/30/2016 09:00:00 AM

This schema defines an xml format for inputing a set of OVAL Definition ids into the reference OVAL Interpreter for evaluation.

< evalutation-definition-ids >

The evaluation-definition-ids element is the root the Document. Its purpose is to bind together the a set of definition elements.

Child Elements

Table 163: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
definition	oval:DefinitionIDPattern (1..unbounded)	Each definition represents the id of a definition to be evaluated.

Open Vulnerability and Assessment Language: Core Variable

- Schema: Core Variable
- Version: 5.11.2
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the core schema for encoding Open Vulnerability and Assessment Language (OVAL) Variables. This schema is provided to give structure to any external variables and their values that an OVAL Definition is expecting.

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_variables >

The oval_variables element is the root of an OVAL Variable Document. Its purpose is to bind together the different variables contained in the document. The generator section must be present and provides information about when the variable file was compiled and under what version. 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. More information about the official W3C Recommendation regarding XML digital signatures can be found at http://www.w3.org/TR/xmldsig-core/.

Child Elements

Table 164: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
generator	oval:GeneratorType (1..1)	
variables	oval-var:VariablesType (0..1)	
ds:Signature	n/a (0..1)	

== VariablesType ==

The VariablesType complex type is a container for one or more variable elements. Each variable element holds the value of an external variable used in an OVAL Definition. Please refer to the description of the VariableType for more information about an individual variable.

Child Elements

Table 165: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
variable	oval-var:VariableType (1..unbounded)	

== VariableType ==

Each variable element contains the associated datatype and value which will be substituted into the OVAL Definition that is referencing this specific variable.

The notes section of a variable should be used to hold information that might be helpful to someone examining the technical aspects of the variable. Please refer to the description of the NotesType complex type for more information about the notes element.

Attributes

Table 166: Attributes

Attribute	Type	Desc.
id	oval:VariableIDPattern (required)	(No Description)
datatype	oval:SimpleDatatypeEnumeration (required)	Note that the ‘record’ datatype is not permitted on variables.
instance	xsd:nonNegativeInteger	Use to specify multiple variable instances.
comment	xsd:string (required)	(No Description)

Child Elements

Table 167: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
value	xsd:anySimpleType (1..unbounded)	
notes	oval:NotesType (0..1)	

Open Vulnerability and Assessment Language: Independent Definition

- Schema: Independent Definition
- Version: 5.11.1:1.2
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the tests found in Open Vulnerability and Assessment Language (OVAL) that are independent of a specific piece of software. Each test is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core Definition Schema is not outlined here.

The OVAL Schema is maintained by the 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.cisecurity.org.

Test Listing

- *< family_test >*
- *< filehash_test > (Deprecated)* (Deprecated)
- *< filehash58_test >*
- *< environmentvariable_test > (Deprecated)* (Deprecated)
- *< environmentvariable58_test >*
- *< ldap_test >*
- *< ldap57_test > (Deprecated)* (Deprecated)
- *< sql_test > (Deprecated)* (Deprecated)
- *< sql57_test >*
- *< textfilecontent54_test >*
- *< textfilecontent_test > (Deprecated)* (Deprecated)
- *< unknown_test >*
- *< variable_test >*
- *< xmlfilecontent_test >*

< family_test >

The family_test element is used to check the family a certain system belongs to. This test basically allows the high level system types (window, unix, ios, etc.) to be tested. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a family_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 168: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< family_object >

The family_object element is used by a family test to define those objects to evaluate based on a specified state. There is actually only one object relating to family and this is the system as a whole. Therefore, there are no child entities defined. Any OVAL Test written to check the family will reference the same family_object which is basically an empty object element.

Extends: oval-def:ObjectType

< family_state >

The family_state element contains a single entity that is used to check the family associated with the system. The family is a high-level classification of system types.

Extends: oval-def:StateType

Child Elements

Table 169: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
family	ind-def:EntityStateFamilyType (0..1)	This element describes the high-level system OS type to test against. Please refer to the definition of the EntityFamilyType for more information about the possible values..

< filehash_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.8
- Reason: Replaced by the filehash58_test.
- Comment: This object has been deprecated and may be removed in a future version of the language.

The file hash test is used to check the hashes associated with a specified file. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a filehash_object and the optional state element specifies the different hashes to check.

Extends: oval-def:TestType

Child Elements

Table 170: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< filehash_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.8
- Reason: Replaced by the filehash58_object.
- Comment: This object has been deprecated and may be removed in a future version of the language.

The filehash_object element is used by a file hash test to define the specific file(s) to be evaluated. The filehash_object will only collect regular files on UNIX systems and FILE_TYPE_DISK files on Windows systems. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A filehash_object defines the path and filename of the file(s). In addition, a number of behaviors may be provided that help guide the collection of objects. Please refer to the FileBehaviors complex type for more information about specific behaviors.

The set of files to be evaluated may be identified with either a complete filepath or a path and filename. Only one of these options may be selected.

It is important to note that 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. This is because the ‘filepath’ element represents an absolute path to a particular file and it is not possible to recurse over a file.

Extends: oval-def:ObjectType

Child Elements

Table 171: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	ind-def:FileBehaviors (0..1)	
filepath	oval-def:EntityObjectStringType (1..1)	The filepath element specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath.
path	oval-def:EntityObjectStringType (1..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-def:EntityObjectStringType (1..1)	The filename element specifies the name of the file.

< filehash_state > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.8
- Reason: Replaced by the filehash58_state.
- Comment: This object has been deprecated and may be removed in a future version of the language.

The filehash_state element contains entities that are used to check the file path, name, and the different hashes associated with a specific file.

Extends: oval-def:StateType

Child Elements

Table 172: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-def:EntityStateStringType (0..1)	The filepath element specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath.
path	oval-def:EntityStateStringType (0..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-def:EntityStateStringType (0..1)	The filename element specifies the name of the file.
md5	oval-def:EntityStateStringType (0..1)	The md5 element is the md5 hash of the file.
sha1	oval-def:EntityStateStringType (0..1)	The sha1 element is the sha1 hash of the file.
windows_view	ind-def:EntityStateWindowsViewType (0..1)	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.

< filehash58_test >

The file hash test is used to check a specific hash type associated with a specified file. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a filehash58_object and the optional state element specifies an expected hash value.

Extends: oval-def:TestType

Child Elements

Table 173: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< filehash58_object >

The filehash58_object element is used by a file hash test to define the specific file(s) to be evaluated. The filehash58_object will only collect regular files on UNIX systems and FILE_TYPE_DISK files on Windows systems. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A filehash58_object defines the path and filename of the file(s). In addition, a number of behaviors may be provided that help guide the collection of objects. Please refer to the FileBehaviors complex type for more information about specific behaviors.

The set of files to be evaluated may be identified with either a complete filepath or a path and filename. Only one of these options may be selected.

It is important to note that 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. This is because the 'filepath' element represents an absolute path to a particular file and it is not possible to recurse over a file.

Extends: oval-def:ObjectType

Child Elements

Table 174: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	ind-def:FileBehaviors (0..1)	
filepath	oval-def:EntityObjectStringType (1..1)	The filepath element specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath.
path	oval-def:EntityObjectStringType (1..1)	The path entity specifies the directory component of the absolute path to a file on the machine.
filename	oval-def:EntityObjectStringType (1..1)	The filename entity specifies the name of the file.
hash_type	ind-def:EntityObjectHashTypeType (1..1)	The hash_type entity specifies the hash algorithm to use when collecting the hash for each of the specifed files.
oval-def:filter	n/a (0..unbounded)	

< filehash58_state >

The filehash58_state element contains entities that are used to check the file path, name, hash_type, and hash associated with a specific file.

Extends: oval-def:StateType

Child Elements

Table 175: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-def:EntityStateStringType (0..1)	The filepath entity specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath.
path	oval-def:EntityStateStringType (0..1)	The path entity specifies the directory component of the absolute path to a file on the machine.
filename	oval-def:EntityStateStringType (0..1)	The filename entity specifies the name of the file.
hash_type	ind-def:EntityStateHashTypeType (0..1)	The hash_type entity specifies the hash algorithm to use when collecting the hash for each of the specifed files.
hash	oval-def:EntityStateStringType (0..1)	The hash entity specifies the result of applying the hash algorithm to the file.
windows_view	ind-def:EntityStateWindowsViewType (0..1)	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.

< environmentvariable_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.8
- Reason: Replaced by the environmentvariable58_test.
- Comment: This object has been deprecated and may be removed in a future version of the language.

The environmentvariable_test element is used to check an environment variable found on the system. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a environmentvariable_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 176: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< environmentvariable_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.8
- Reason: Replaced by the environmentvariable58_object.
- Comment: This object has been deprecated and may be removed in a future version of the language.

The environmentvariable_object element is used by an environment variable test to define the specific environment variable(s) to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 177: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityObjectStringType (1..1)	This element describes the name of an environment variable.

< environmentvariable_state > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.8
- Reason: Replaced by the environmentvariable58_state.
- Comment: This object has been deprecated and may be removed in a future version of the language.

The environmentvariable_state element contains two entities that are used to check the name of the specified environment variable and the value associated with it.

Extends: oval-def:StateType

Child Elements

Table 178: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	This element describes the name of an environment variable.
value	oval-def:EntityStateAnySimpleType (0..1)	The actual value of the specified environment variable.

< environmentvariable58_test >

The environmentvariable58_test element is used to check an environment variable for the specified process, which is identified by its process ID, on the system . It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a environmentvariable_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 179: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< environmentvariable58_object >

The environmentvariable58_object element is used by an environmentvariable58_test to define the specific environment variable(s) and process IDs to be evaluated. If a tool is unable to collect the environment variables of another process, an error must be reported. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 180: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
pid	oval-def:EntityObjectIntType (1..1)	The process ID of the process from which the environment variable should be retrieved. If the xsi:nil attribute is set to true, the process ID shall be the tool's running process; for scanners with no process ID (e.g., an agentless network scanner), no corresponding items will exist.
name	oval-def:EntityObjectStringType (1..1)	This element describes the name of an environment variable.
oval-def:filter	n/a (0..unbounded)	

< environmentvariable58_state >

The environmentvariable58_state element contains three entities that are used to check the name of the specified environment variable, the process ID of the process from which the environment variable was retrieved, and the value associated with the environment variable.

Extends: oval-def:StateType

Child Elements

Table 181: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
pid	oval-def:EntityStateIntType (0..1)	The process ID of the process from which the environment variable was retrieved.
name	oval-def:EntityStateStringType (0..1)	This element describes the name of an environment variable.
value	oval-def:EntityStateAnySimpleType (0..1)	The actual value of the specified environment variable.

< ldap_test >

The LDAP test is used to check information about specific entries in an LDAP directory. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an ldap_object and the optional state element, ldap_state, specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 182: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< ldap_object >

The ldap_object element is used by an LDAP test to define the objects to be evaluated based on a specified state. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 183: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	ind-def:LdapBehaviors (0..1)	
suffix	oval-def:EntityObjectStringType (1..1)	Each object in an LDAP directory exists under a certain suffix (also known as a naming context). A suffix is defined as a single object in the Directory Information Tree (DIT) with every object in the tree subordinate to it.
relative_dn	oval-def:EntityObjectStringType (1..1)	The relative_dn field is used to uniquely identify an object inside the specified suffix. It contains all of the parts of the object's distinguished name except those outlined by the suffix. If the xsi:nil attribute is set to true, then the object being specified is the higher level suffix. In this case, the relative_dn element should not be collected or used in analysis. Setting xsi:nil equal to true is different than using a .* pattern match, which says to collect every relative distinguished name under a given suffix.
attribute	oval-def:EntityObjectStringType (1..1)	Specifies a named value contained by the object. If the xsi:nil attribute is set to true, the attribute element should not be collected or used in analysis. Setting xsi:nil equal to true is different than using a .* pattern match, which says to collect every attribute under a given relative distinguished name.

< ldap_state >

The ldap_state element defines the different information that can be used to evaluate the specified entries in an LDAP directory. An ldap_test will reference a specific instance of this state that defines the exact settings that need to be evaluated. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 184: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
suffix	oval-def:EntityStateStringType (0..1)	Each object in an LDAP directory exists under a certain suffix (also known as a naming context). A suffix is defined as a single object in the Directory Information Tree (DIT) with every object in the tree subordinate to it.
relative_dn	oval-def:EntityStateStringType (0..1)	The relative_dn field is used to uniquely identify an object inside the specified suffix. It contains all of the parts of the object's distinguished name except those outlined by the suffix.
attribute	oval-def:EntityStateStringType (0..1)	Specifies a named value contained by the object.
object_class	oval-def:EntityStateStringType (0..1)	The name of the class of which the object is an instance.
ldaptype	ind-def:EntityStateLdaptypeType (0..1)	Specifies the type of information that the specified attribute represents.
value	oval-def:EntityStateAnySimpleType (0..1)	The actual value of the specified LDAP attribute.

== LdapBehaviors ==

The LdapBehaviors complex type defines a number of behaviors that allow a more detailed definition of the ldap_object being specified.

Attributes

Table 185: Attributes

Attribute	Type	Desc.
scope	Restriction of xsd:string (optional *default*='BASE') ('BASE', 'ONE', 'SUBTREE')	'scope' defines the depth from the base distinguished name to which the search should occur. The base distinguished name is the starting point of the search and is composed of the specified suffix and relative distinguished name. A value of 'BASE' indicates to search only the entry at the base distinguished name, a value of 'ONE' indicates to search all entries one level under the base distinguished name - but NOT including the base distinguished name, and a value of 'SUBTREE' indicates to search all entries at all levels under, and including, the specified base distinguished name. The default value is 'BASE'.

< ldap57_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.11.2
- Reason: Use the original ldap_test. The ldap57_test suffers from ambiguity; it was never adequately specified, and it does not even seem possible to have structured data in the context of the enumerated LdaptypeTypes. Use the original ldap_test instead.
- Comment: This test has been deprecated and will be removed in version 6.0 of the language.

The LDAP test is used to check information about specific entries in an LDAP directory. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an ldap57_object and the optional state element, ldap57_state, specifies the metadata to check.

Note that this test supports complex values that are in the form of a record. For simple (string based) value collection see the ldap_test.

Extends: oval-def:TestType

Child Elements

Table 186: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< ldap57_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.11.2
- Reason: Use the original ldap_object. The ldap57_test suffers from ambiguity; it was never adequately specified, and it does not even seem possible to have structured data in the context of the enumerated LdaptypeTypes. Use the original ldap_test instead.
- Comment: This test has been deprecated and will be removed in version 6.0 of the language.

The ldap57_object element is used by an LDAP test to define the objects to be evaluated based on a specified state. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Note that this object supports complex values that are in the form of a record. For simple (string based) value collection see the ldap_object.

Extends: oval-def:ObjectType

Child Elements

Table 187: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	ind-def:LdapBehaviors (0..1)	
suffix	oval-def:EntityObjectStringType (1..1)	Each object in an LDAP directory exists under a certain suffix (also known as a naming context). A suffix is defined as a single object in the Directory Information Tree (DIT) with every object in the tree subordinate to it.
relative_dn	oval-def:EntityObjectStringType (1..1)	The relative_dn field is used to uniquely identify an object inside the specified suffix. It contains all of the parts of the object's distinguished name except those outlined by the suffix. If the xsi:nil attribute is set to true, then the object being specified is the higher level suffix. In this case, the relative_dn element should not be collected or used in analysis. Setting xsi:nil equal to true is different than using a .* pattern match, which says to collect every relative distinguished name under a given suffix.
attribute	oval-def:EntityObjectStringType (1..1)	Specifies a named value contained by the object. If the xsi:nil attribute is set to true, the attribute element should not be collected or used in analysis. Setting xsi:nil equal to true is different than using a .* pattern match, which says to collect every attribute under a given relative distinguished name.
oval-def:filter	n/a (0..unbounded)	

< ldap57_state > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.11.2
- Reason: Use the original ldap_state. The ldap57_test suffers from ambiguity; it was never adequately specified, and it does not even seem possible to have structured data in the context of the enumerated LdaptypeTypes. Use the original ldap_test instead.
- Comment: This test has been deprecated and will be removed in version 6.0 of the language.

The ldap57_state element defines the different information that can be used to evaluate the specified entries in an LDAP directory. An ldap57_test will reference a specific instance of this state that defines the exact settings that need to be evaluated. Please refer to the individual elements in the schema for more details about what each represents.

Note that this state supports complex values that are in the form of a record. For simple (string based) value collection see the ldap_state.

Extends: oval-def:StateType

Child Elements

Table 188: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
suffix	oval-def:EntityStateStringType (0..1)	Each object in an LDAP directory exists under a certain suffix (also known as a naming context). A suffix is defined as a single object in the Directory Information Tree (DIT) with every object in the tree subordinate to it.
relative_dn	oval-def:EntityStateStringType (0..1)	The relative_dn field is used to uniquely identify an object inside the specified suffix. It contains all of the parts of the object's distinguished name except those outlined by the suffix.
attribute	oval-def:EntityStateStringType (0..1)	Specifies a named value contained by the object.
object_class	oval-def:EntityStateStringType (0..1)	The name of the class of which the object is an instance.
ldaptype	ind-def:EntityStateLdaptypeType (0..1)	Specifies the type of information that the specified attribute represents.
value	oval-def:EntityStateRecordType (0..1)	The actual value of the specified LDAP attribute. Note that while an LDAP attribute can contain structured data where it is necessary to collect multiple related fields that can be described by the 'record' datatype, it is not always the case. It also is possible that an LDAP attribute can contain only a single value or an array of values. In these cases, there is not a name to uniquely identify the corresponding field which is a requirement for fields in the 'record' datatype. As a result, the name of the LDAP attribute will be used to uniquely identify the field and satisfy this requirement.

< sql_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.7
- Reason: Replaced by the sql57_test. This test allows for single fields to be selected from a database. A new test was created to allow more than one field to be selected in one statement. See the sql57_test.
- Comment: This object has been deprecated and may be removed in a future version of the language.

The sql test is used to check information stored in a database. It is often the case that applications store configuration settings in a database as opposed to a file. This test has been designed to enable those settings to be tested. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a wmi_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 189: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< sql_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.7
- Reason: Replaced by the sql57_object. This object allows for single fields to be selected from a database. A new object was created to allow more than one field to be selected in one statement. See the sql57_object.
- Comment: This object has been deprecated and may be removed in a future version of the language.

The sql_object element is used by a sql test to define the specific database and query to be evaluated. Connection information is supplied allowing the tool to connect to the desired database and a query is supplied to call out the desired setting. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 190: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
engine	ind-def:EntityObjectEngineType (1..1)	The engine entity defines the specific database engine to use. Any tool looking to collect information about this object will need to know the engine in order to use the appropriate drivers to establish a connection.
version	oval-def:EntityObjectStringType (1..1)	The version entity defines the specific version of the database engine to use. This is also important in determining the correct driver to use for establishing a connection.
connection_string	oval-def:EntityObjectStringType (1..1)	The connection_string entity defines specific connection parameters to be used in connecting to the database. This will help a tool connect to the correct database.
sql	oval-def:EntityObjectStringType (1..1)	The sql entity defines a query used to identify the object(s) to test against. Any valid SQL query is usable with one exception, at most one field is allowed in the SELECT portion of the query. For example SELECT name FROM … is valid, as is SELECT 'true' FROM …, but SELECT name, number FROM … is not valid. This is because the result element in the data section is only designed to work against a single field.

< sql_state > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.7
- Reason: Replaced by the sql57_state. This state allows for single fields to be selected from a database. A new state was created to allow more than one field to be selected in one statement. See the sql57_state.
- Comment: This state has been deprecated and may be removed in a future version of the language.

The sql_state element contains two entities that are used to check the name of the specified field and the value associated with it.

Extends: oval-def:StateType

Child Elements

Table 191: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
engine	ind-def:EntityStateEngineType (0..1)	The engine entity defines a specific database engine.
version	oval-def:EntityStateStringType (0..1)	The version entity defines a specific version of a given database engine.
connection_string	oval-def:EntityStateStringType (0..1)	The connection_string entity defines a set of parameters that help identify the connection to the database.
sql	oval-def:EntityStateStringType (0..1)	the sql entity defines a query used to identify the object(s) to test against.
result	oval-def:EntityStateAnySimpleType (0..1)	The result entity specifies how to test objects in the result set of the specified SQL statement. Only one comparable field is allowed. So if the SQL statement look like 'SELECT name FROM ...', then a result entity with a value of 'Fred' would test the set of 'name' values returned by the SQL statement against the value 'Fred'.

< sql57_test >

The sql test is used to check information stored in a database. It is often the case that applications store configuration settings in a database as opposed to a file. This test has been designed to enable those settings to be tested. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a wmi_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 192: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< sql57_object >

The sql57_object element is used by a sql test to define the specific database and query to be evaluated. Connection information is supplied allowing the tool to connect to the desired database and a query is supplied to call out the desired setting. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 193: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
engine	ind-def:EntityObjectEngineType (1..1)	The engine entity defines the specific database engine to use. Any tool looking to collect information about this object will need to know the engine in order to use the appropriate drivers to establish a connection.
version	oval-def:EntityObjectStringType (1..1)	The version entity defines the specific version of the database engine to use. This is also important in determining the correct driver to use for establishing a connection.
connection_string	oval-def:EntityObjectStringType (1..1)	The connection_string entity defines specific connection parameters to be used in connecting to the database. This will help a tool connect to the correct database.
sql	oval-def:EntityObjectStringType (1..1)	The sql entity defines a query used to identify the object(s) to test against. Any valid SQL query is usable with one exception, all fields must be named in the SELECT portion of the query. For example, SELECT name, number FROM … is valid. However, SELECT * FROM … is not valid. This is because the record element in the state and item require a unique field name value to ensure that any query results can be evaluated consistently.
oval-def:filter	n/a (0..unbounded)	

< sql57_state >

The sql57_state element contains two entities that are used to check the name of the specified field and the value associated with it.

Extends: oval-def:StateType

Child Elements

Table 194: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
engine	ind-def:EntityStateEngineType (0..1)	The engine entity defines a specific database engine.
version	oval-def:EntityStateStringType (0..1)	The version entity defines a specific version of a given database engine.
connection_string	oval-def:EntityStateStringType (0..1)	The connection_string entity defines a set of parameters that help identify the connection to the database.
sql	oval-def:EntityStateStringType (0..1)	the sql entity defines a query used to identify the object(s) to test against.
result	oval-def:EntityStateRecordType (0..1)	The result entity specifies how to test objects in the result set of the specified SQL statement.

< textfilecontent54_test >

The textfilecontent54_test element is used to check the contents of a text file (aka a configuration file) by looking at individual blocks of text. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a textfilecontent54_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 195: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< textfilecontent54_object >

The textfilecontent54_object element is used by a textfilecontent_test to define the specific block(s) of text of a file(s) to be evaluated. The textfilecontent54_object will only collect regular files on UNIX systems and FILE_TYPE_DISK files on Windows systems. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

The set of files to be evaluated may be identified with either a complete filepath or a path and filename. Only one of these options may be selected.

It is important to note that 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. This is because the ‘filepath’ element represents an absolute path to a particular file and it is not possible to recurse over a file.

Extends: oval-def:ObjectType

Child Elements

Table 196: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	ind-def:Textfilecontent54Behaviors (0..1)	
filepath	oval-def:EntityObjectStringType (1..1)	The filepath element specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath.
path	oval-def:EntityObjectStringType (1..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-def:EntityObjectStringType (1..1)	The filename entity specifies the name of a file.
pattern	oval-def:EntityObjectStringType (1..1)	The pattern entity defines a chunk of text in a file and is 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. For more information on the supported regular expression syntax in OVAL see: http://oval.mitre.org/language/about/re_support_5.6.html.
instance	oval-def:EntityObjectIntType (1..1)	The instance entity calls out a specific match of the pattern. It can have both positive and negative values. If the value is positive, the index of the specific match of the pattern is counted from the beginning of the set of matches of that pattern. The first match is given an instance value of 1, the second match is given an instance value of 2, and so on. For positive values, the 'less than' and 'less than or equals' operations imply the the object is operating only on positive values. Frequently, this entity will be defined as 'greater than or equals' 1, which results in the object representing the set of all matches of the pattern.Negative values are used to simplify collection of pattern match occurrences counting backwards from the last match. To find the last match, use an instance of -1; the penultimate match is found using an instance value of -2, and so on. For negative values, the 'greater than' and 'greater than or equals' operations imply the object is operating only on negative values. For example, searching for instances greater than or equal to -2 would yield only the last two maches.Note that the main purpose of the instance item entity is to provide uniqueness for different textfilecontent_items that results from multiple matches of a given pattern against the same file, and they will always have positive values.
oval-def:filter	n/a (0..unbounded)	

< textfilecontent54_state >

The textfilecontent54_state element contains entities that are used to check the file path and name, as well as the text block in question and the value of the subexpressions.

Extends: oval-def:StateType

Child Elements

Table 197: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-def:EntityStateStringType (0..1)	The filepath element specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath.
path	oval-def:EntityStateStringType (0..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-def:EntityStateStringType (0..1)	The filename entity represents the name of a file.
pattern	oval-def:EntityStateStringType (0..1)	The pattern entity represents a regular expression that is used to define a block of text.
instance	oval-def:EntityStateIntType (0..1)	The instance entity calls out a specific match of the pattern. This can only be a positive integer.
text	oval-def:EntityStateAnySimpleType (0..1)	The text entity represents the block of text that matched the specified pattern.
subexpression	oval-def:EntityStateAnySimpleType (0..1)	The subexpression entity represents a value to test against the subexpression in the specified pattern. If multiple subexpressions are specified in the pattern, this value is tested against all of them. For example, if the pattern abc(.*)mno(.*)xyp was supplied, and the state specifies a subexpression value of enabled, then the test would check that both (or at least one, none, etc. depending on the entity_check attribute) of the subexpressions have a value of enabled.
windows_view	ind-def:EntityStateWindowsViewType (0..1)	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.

== Textfilecontent54Behaviors ==

The Textfilecontent54Behaviors complex type defines a number of behaviors that allow a more detailed definition of the textfilecontent54_object being specified. 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.

It is important to note that 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. This is because the ‘filepath’ element represents an absolute path to a particular file and it is not possible to recurse over a file.

The Textfilecontent54Behaviors extend the ind-def:FileBehaviors and therefore include the behaviors defined by that type.

Extends: ind-def:FileBehaviors

Attributes

Table 198: Attributes

Attribute	Type	Desc.
ignore_case	xsd:boolean (optional *default*='false')	'ignore_case' 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: if true, case will be ignored. If false, case will not be ignored. The default is false.
multiline	xsd:boolean (optional *default*='true')	'multiline' 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: if true, the '^' and '$' metacharacters will match both at the beginning/end of a string, and immediately after/before newline characters. If false, they will match only at the beginning/end of a string. The default is true.
singleline	xsd:boolean (optional *default*='false')	'singleline' enables single line semantics in the regular expression provided by the pattern entity. This behavior is intended to align with the Perl regular expression 's' modifier: if true, the '.' metacharacter will match newlines. If false, it will not. The default is false.

< textfilecontent_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.4
- Reason: Replaced by the textfilecontent54_test. Support for multi-line pattern matching and multi-instance matching was added. Therefore, a new test was created to reflect these changes. See the textfilecontent54_test.
- Comment: This test has been deprecated and will be removed in version 6.0 of the language.

The textfilecontent_test element is used to check the contents of a text file (aka a configuration file) by looking at individual lines. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a textfilecontent_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 199: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< textfilecontent_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.4
- Reason: Replaced by the textfilecontent54_object. Support for multi-line pattern matching and multi-instance matching was added. Therefore, a new object was created to reflect these changes. See the textfilecontent54_object.
- Comment: This object has been deprecated and will be removed in version 6.0 of the language.

The textfilecontent_object element is used by a text file content test to define the specific line(s) of a file(s) to be evaluated. The textfilecontent_object will only collect regular files on UNIX systems and FILE_TYPE_DISK files on Windows systems. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 200: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	ind-def:FileBehaviors (0..1)	
path	oval-def:EntityObjectStringType (1..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-def:EntityObjectStringType (1..1)	The filename element specifies the name of the file.
line	oval-def:EntityObjectStringType (1..1)	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. The value of this subexpression can then be checked using a textfilecontent_state.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. For more information on the supported regular expression syntax in OVAL see: http://oval.mitre.org/language/about/re_support_5.6.html.

< textfilecontent_state > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.4

- Reason: Replaced by the textfilecontent54_state. Support for multi-line pattern matching and multi-instance matching was added. Therefore, a new state was created to reflect these changes. See the textfilecontent54_state.
- Comment: This state has been deprecated and will be removed in version 6.0 of the language.

The textfilecontent_state element contains entities that are used to check the file path and name, as well as the line in question and the value of the specific subexpression.

Extends: oval-def:StateType

Child Elements

Table 201: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
path	oval-def:EntityStateStringType (0..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-def:EntityStateStringType (0..1)	The name of the file.
line	oval-def:EntityStateStringType (0..1)	The line element represents a line in the file that was collected.
subexpression	oval-def:EntityStateAnySimpleType (0..1)	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)	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.

< unknown_test >

An unknown_test acts as a placeholder for tests whose implementation is unknown. This test always evaluates to a result of ‘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. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. Note that for an unknown_test, 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.

Extends: oval-def:TestType

< variable_test >

The variable test allows the value of a variable to be compared to a defined value. As an example one might use this test to validate that a variable being passed in from an external source falls within a specified range. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a variable_object and the optional state element specifies the value to check.

Extends: oval-def:TestType

Child Elements

Table 202: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< variable_object >

Extends: oval-def:ObjectType

Child Elements

Table 203: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
var_ref	ind-def:EntityObjectVariableRefType (1..1)	The id of the variable you want.
oval-def:filter	n/a (0..unbounded)	

< variable_state >

The variable_state element contains two entities that are used to check the var_ref of the specified varible and the value associated with it.

Extends: oval-def:StateType

Child Elements

Table 204: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
var_ref	ind-def:EntityStateVariableRefType (0..1)	The id of the variable.
value	oval-def:EntityStateAnySimpleType (0..1)	The value of the variable.

< xmlfilecontent_test >

The xmlfilecontent_test element is used to explore the contents of an xml file. This test allows specific pieces of an xml document specified using xpath to be tested. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a xmlfilecontent_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 205: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< xmlfilecontent_object >

The xmlfilecontent_object element is used by a xml file content test to define the specific piece of an xml file(s) to be evaluated. The xmlfilecontent_object will only collect regular files on UNIX systems and FILE_TYPE_DISK files on Windows systems. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

The set of files to be evaluated may be identified with either a complete filepath or a path and filename. Only one of these options may be selected.

It is important to note that 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. This is because the 'filepath' element represents an absolute path to a particular file and it is not possible to recurse over a file.

Extends: oval-def:ObjectType

Child Elements

Table 206: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	ind-def:FileBehaviors (0..1)	
filepath	oval-def:EntityObjectStringType (1..1)	The filepath element specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath.
path	oval-def:EntityObjectStringType (1..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-def:EntityObjectStringType (1..1)	The filename element specifies the name of the file.
xpath	oval-def:EntityObjectStringType (1..1)	Specifies an XPath 1.0 expression to evaluate against the XML file specified by the filename entity. This XPath 1.0 expression must evaluate to a list of zero or more text values which will be accessible in OVAL via instances of the value_of entity. Any results from evaluating the XPath 1.0 expression other than a list of text strings (e.g., a nodes set) is considered an error. The intention is that the text values be drawn from instances of a single, uniquely named element or attribute. However, an OVAL interpreter is not required to verify this, so the author should define the XPath expression carefully. Note that “equals” is the only valid operator for the xpath entity.
oval-def:filter	n/a (0..unbounded)	

< xmlfilecontent_state >

The xmlfilecontent_state element contains entities that are used to check the file path and name, as well as the xpath used and the value of the this xpath.

Extends: oval-def:StateType

Child Elements

Table 207: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-def:EntityStateStringType (0..1)	The filepath element specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath.
path	oval-def:EntityStateStringType (0..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-def:EntityStateStringType (0..1)	The filename element specifies the name of the file.
xpath	oval-def:EntityStateStringType (0..1)	Specifies an XPath 1.0 expression to evaluate against the XML file specified by the filename entity. This XPath 1.0 expression must evaluate to a list of zero or more text values which will be accessible in OVAL via instances of the value_of entity. Any results from evaluating the XPath 1.0 expression other than a list of text strings (e.g., a nodes set) is considered an error. The intention is that the text values be drawn from instances of a single, uniquely named element or attribute. However, an OVAL interpreter is not required to verify this, so the author should define the XPath expression carefully. Note that "equals" is the only valid operator for the xpath entity.
value_of	oval-def:EntityStateAnySimpleType (0..1)	The value_of element checks the value(s) of the text node(s) or attribute(s) found.
windows_view	ind-def:EntityStateWindowsViewType (0..1)	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.

== FileBehaviors ==

The FileBehaviors complex type defines a number of behaviors that allow a more detailed definition of a set of files or file related items to collect. 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.

It is important to note that 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. This is because the 'filepath' element represents an absolute path to a particular file and it is not possible to recurse over a file.

Attributes

Table 208: Attributes

Attribute	Type	Desc.
max_depth	Restriction of xsd:integer (optional *default*='-1')	'max_depth' defines the maximum depth of recursion to perform when a recurse_direction is specified. A value of '0' is equivalent to no recursion, '1' means to step only one directory level up/down, and so on. The default value is '-1' meaning no limitation. For a 'max_depth' of -1 or any value of 1 or more the starting directory must be considered in the recursive search.

Note that the default recurse_direction behavior is 'none' so even though max_depth specifies no limitation by default, the recurse_direction behavior turns recursion off. Note that this behavior only applies with the equality operation on the path entity.

- - recurse
  - Restriction of xsd:string (optional *default*='symlinks and directories') ('directories', 'symlinks', 'symlinks and directories')
  - 'recurse' defines how to recurse into the path entity, in other words what to follow during recursion. Options include symlinks, directories, or both. Note that a max-depth other than 0 has to be specified for recursion to take place and for this attribute to mean anything. Also note that on Windows, the 'symlink' value is equivalent to the 'junction' recurse value in win-def:FileBehaviors.

Note that this behavior only applies with the equality operation on the path entity.

- - recurse_direction
  - Restriction of xsd:string (optional *default*='none') ('none', 'up', 'down')
  - 'recurse_direction' defines the direction to recurse, either 'up' to parent directories, or 'down' into child directories. The default value is 'none' for no recursion.

Note that this behavior only applies with the equality operation on the path entity.

- - recurse_file_system
  - Restriction of xsd:string (optional *default*='all') ('all', 'local', 'defined')
  - 'recurse_file_system' defines the file system limitation of any searching and applies to all operations as specified on the path or filepath entity. The value of 'local' limits the search scope to local file systems (as opposed to file systems mounted from an external system). The value of 'defined' keeps any recursion within the file system that the file_object (path+filename or filepath) has specified. For example, on Windows, if the path specified was "C:", you would search only the C: drive, not other filesystems mounted to descendant paths. Similarly, on UNIX, if the path specified was "/", you would search only the filesystem mounted there, not other filesystems mounted to descendant paths. The value of 'defined' only applies when an equality operation is used for searching because the path or filepath entity must explicitly define a file system. The default value is 'all' meaning to search all available file systems for data collection.

Note that in most cases it is recommended that the value of 'local' be used to ensure that file system searching is limited to only t

- - windows_view
  - Restriction of xsd:string (optional *default*='64_bit') ('32_bit', '64_bit')

  - 64-bit versions of Windows provide an alternate file system and registry views to 32-bit applications. This behavior allows the OVAL Object to specify which view should be examined. This behavior only applies to 64-bit Windows, and must not be applied on other platforms.

Note that the values have the following meaning: ‘64_bit’ – Indicates that the 64-bit view on 64-bit Windows operating systems must be examined. On a 32-bit system, the Object must be evaluated without applying the behavior. ‘32_bit’ – Indicates that the 32-bit view must be examined. On a 32-bit system, the Object must be evaluated without applying the behavior. It is recommended that the corresponding ‘windows_view’ entity be set on the OVAL Items that are collected when this behavior is used to distinguish between the OVAL Items that are collected in the 32-bit or 64-bit views.

== EntityObjectEngineType ==

The EntityObjectEngineType complex type defines a string entity value that is restricted to a set of enumerations. Each valid enumeration is a valid database engine. The empty string is also allowed to support empty elements associated with variable references.

Restricts: oval-def:EntityObjectStringType

Table 209: Enumeration Values

Value	Description
access	The access value describes the Microsoft Access database engine.
db2	The db2 value describes the IBM DB2 database engine.
cache	The cache value describes the InterSystems Cache database engine.
firebird	The firebird value describes the Firebird database engine.
firstsql	The firstsql value describes the FirstSQL database engine.
foxpro	The foxpro value describes the Microsoft FoxPro database engine.
informix	The informix value describes the IBM Informix database engine.
ingres	The ingres value describes the Ingres database engine.
interbase	The interbase value describes the Embarcadero Technologies InterBase database engine.
lightbase	The lightbase value describes the Light Infocon LightBase database engine.
maxdb	The maxdb value describes the SAP MaxDB database engine.
monetdb	The monetdb value describes the MonetDB SQL database engine.
mimer	The mimer value describes the Mimer SQL database engine.

== EntityStateEngineType ==

The EntityStateEngineType complex type defines a string entity value that is restricted to a set of enumerations. Each valid enumeration is a valid database engine. The empty string is also allowed to support empty elements associated with variable references.

Restricts: oval-def:EntityStateStringType

Table 210: Enumeration Values

Value	Description
access	The access value describes the Microsoft Access database engine.
db2	The db2 value describes the IBM DB2 database engine.
cache	The cache value describes the InterSystems Cache database engine.
firebird	The firebird value describes the Firebird database engine.
firstsql	The firstsql value describes the FirstSQL database engine.
foxpro	The foxpro value describes the Microsoft FoxPro database engine.
informix	The informix value describes the IBM Informix database engine.
ingres	The ingres value describes the Ingres database engine.
interbase	The interbase value describes the Embarcadero Technologies InterBase database engine.
lightbase	The lightbase value describes the Light Infocon LightBase database engine.
maxdb	The maxdb value describes the SAP MaxDB database engine.
monetdb	The monetdb value describes the MonetDB SQL database engine.
mimer	The mimer value describes the Mimer SQL database engine.

== EntityStateFamilyType ==

The EntityStateFamilyType complex type defines a string entity value that is restricted to a set of enumerations. Each valid enumeration is a high-level family of system operating system. The empty string is also allowed to support empty elements associated with variable references.

Restricts: oval-def:EntityStateStringType

Table 211: Enumeration Values

Value	Description
android	The android value describes the Android mobile operating system.
apple_ios	The apple_ios value describes the iOS mobile operating system.
asa	The asa value describes the Cisco ASA security devices.
catos	The catos value describes the Cisco CatOS operating system.
ios	The ios value describes the Cisco IOS operating system.
iosxe	The iosxe value describes the Cisco IOS-XE operating system.
junos	The junos value describes the Juniper JunOS operating system.
macos	The macos value describes the Mac operating system.
pixos	The pixos value describes the Cisco PIX operating system.
undefined	The undefined value is to be used when the desired family is not available.
unix	The unix value describes the UNIX operating system.
vmware_infrastructure	The vmware_infrastructure value describes VMWare Infrastructure.
windows	The windows value describes the Microsoft Windows operating system.

== EntityObjectHashTypeType ==

The EntityObjectHashTypeType complex type restricts a string value to a specific set of values that specify the different hash algorithms that are supported. The empty string is also allowed to support empty elements associated with variable references.

Restricts: oval-def:EntityObjectStringType

Table 212: Enumeration Values

Value	Description
MD5	The MD5 hash algorithm.
SHA-1	The SHA-1 hash algorithm.
SHA-224	The SHA-224 hash algorithm.
SHA-256	The SHA-256 hash algorithm.
SHA-384	The SHA-384 hash algorithm.
SHA-512	The SHA-512 hash algorithm.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateHashTypeType ==

The EntityStateHashTypeType complex type restricts a string value to a specific set of values that specify the different hash algorithms that are supported. The empty string is also allowed to support empty elements associated with variable references.

Restricts: oval-def:EntityStateStringType

Table 213: Enumeration Values

Value	Description
MD5	The MD5 hash algorithm.
SHA-1	The SHA-1 hash algorithm.
SHA-224	The SHA-224 hash algorithm.
SHA-256	The SHA-256 hash algorithm.
SHA-384	The SHA-384 hash algorithm.
SHA-512	The SHA-512 hash algorithm.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityObjectVariableRefType ==

The EntityObjectVariableRefType complex type defines a string object entity that has a valid OVAL variable id as the value. The empty string is also allowed to support empty elements associated with variable references.

Restricts: oval-def:EntityObjectStringType

Pattern: (oval:[**A-Za-z0-9_**-.]+:var:[1-9][0-9]*){0,}

== EntityStateVariableRefType ==

The EntityStateVariableRefType complex type defines a string state entity that has a valid OVAL variable id as the value. The empty string is also allowed to support empty elements associated with variable references.

Restricts: oval-def:EntityStateStringType

Pattern: (oval:[**A-Za-z0-9_**-.]+:var:[1-9][0-9]*){0,}

== EntityStateLdaptypeType ==

The EntityStateLdaptypeType complex type restricts a string value to a specific set of values that specify the different types of information that an ldap attribute can represent. The empty string is also allowed to support empty elements associated with variable references.

Restricts: oval-def:EntityStateStringType

Table 214: Enumeration Values

Value	Description
LDAPTYPE_ACI_ITEM	ACI Item, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.1
LDAPTYPE_ACCESS_POINT	Access Point, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.2
LDAPTYPE_ATTRIBUTE_TYPE_DESCRIP_STRING	Attribute Type Description, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.3
LDAPTYPE_AUDIO	Audio, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.4
LDAPTYPE_BINARY	Binary, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.5
LDAPTYPE_BIT_STRING	Bit String, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.6
LDAPTYPE_BOOLEAN	Boolean, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.7
LDAPTYPE_CERTIFICATE	Certificate, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.8
LDAPTYPE_CERTIFICATE_LIST	Certificate List, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.9
LDAPTYPE_CERTIFICATE_PAIR	Certificate Pair, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.10
LDAPTYPE_COUNTRY_STRING	Country String, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.11
LDAPTYPE_DN_STRING	DN, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.12

Continued on next page

Table 214 – continued from previous page

Value	Description
LDAPTYPE_DATA_QUALITY_SYNTAX	Data Quality Syntax, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.13
LDAPTYPE_DELIVERY_METHOD	Delivery Method, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.14
LDAPTYPE_DIRECTORY_STRING	Directory String, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.15
LDAPTYPE_DIR_CONTENT_RULE_DESCRIPTION	DIT Content Rule Description, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.16
LDAPTYPE_DIT_STRUCTURE_RULE_DESCRIPTION	DIT Structure Rule Description, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.17
LDAPTYPE_DL_SUBMIT_PERMISSION	DL Submit Permission, corresponding to OID Y 1.3.6.1.4.1.1466.115.121.1.18
LDAPTYPE_DSA_QUALITY_SYNTAX	DSA Quality Syntax, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.19
LDAPTYPE_DSE_TYPE	DSE Type, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.20
LDAPTYPE_ENHANCED_GUIDE	Enhanced Guide, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.21
LDAPTYPE_FAX_TEL_NUMBER	Facsimile Telephone Number, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.22
LDAPTYPE_FAX	Fax, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.23
LDAPTYPE_GENERALIZED_TIME	Generalized Time, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.24

Continued on next page

Table 214 – continued from previous page

Value	Description
LDAPTYPE_GUIDE	Guide, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.25
LDAPTYPE_IA5_STRING	IA5 String, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.26
LDAPTYPE_INTEGER	INTEGER, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.27
LDAPTYPE_JPEG	JPEG, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.28
LDAPTYPE_LDAP_SYNTAX_DESCRIPTION	LDAP Syntax Description, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.54
LDAPTYPE_LDAP_SCHEMA_DEFINITION	LDAP Schema Definition, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.56
LDAPTYPE_LDAP_SCHEMA_DESCRIPTION	LDAP Schema Description, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.57
LDAPTYPE_MASTER_AND_SHADOW_ACCESS_POINTS	Master And Shadow Access Points, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.29
LDAPTYPE_MATCHING_RULE_DESCRIPTION	Matching Rule Description, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.30
LDAPTYPE_MATCHING_RULE_USE_DESCRIPTION	Matching Rule Use Description, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.31
LDAPTYPE_MAIL_PREFERENCE	Mail Preference, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.32
LDAPTYPE_MHS_OR_ADDRESS	MHS OR Address, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.33

Continued on next page

Table 214 – continued from previous page

Value	Description
LDAPTYPE_MODIFY_RIGHTS	Modify Rights, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.55
LDAPTYPE_NAME_AND_OPTIONAL_UID	Name And Optional UID, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.34
LDAPTYPE_NAME_FORM_DESCRIPTION	Name Form Description, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.35
LDAPTYPE_NUMERIC_STRING	Numeric String, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.36
LDAPTYPE_OBJECT_CLASS_DESCRIP_STRING	Object Class Description, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.37
LDAPTYPE_OCTET_STRING	Octet String, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.40
LDAPTYPE_OID	OID, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.38
LDAPTYPE_MAILBOX	Other Mailbox, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.39
LDAPTYPE_POSTAL_ADDRESS	Postal Address, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.41
LDAPTYPE_PROTOCOL_INFORMATION	Protocol Information, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.42
LDAPTYPE_PRESENTATION_ADDRESS	Presentation Address, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.43
LDAPTYPE_PRINTABLE_STRING	Printable String, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.44

Continued on next page

Table 214 – continued from previous page

Value	Description
LDAPTYPE_SUBSTRING_ASSERTION	Substring Assertion, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.58
LDAPTYPE_SUBTREE_SPECIFICATION	Subtree Specification, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.45
LDAPTYPE_SUPPLIER_INFORMATION	Supplier Information, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.46
LDAPTYPE_SUPPLIER_OR_CONSUMER	Supplier Or Consumer, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.47
LDAPTYPE_SUPPLIER_AND_CONSUMER	Supplier And Consumer, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.48
LDAPTYPE_SUPPORTED_ALGORITHM	Supported Algorithm, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.49
LDAPTYPE_TELEPHONE_NUMBER	Telephone Number, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.50
LDAPTYPE_TELEX_TERMINAL_ID	Teletex Terminal Identifier, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.51
LDAPTYPE_TELEX_NUMBER	Telex Number, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.52
LDAPTYPE_UTC_TIME	UTC Time, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.53
LDAPTYPE_TIMESTAMP (Deprecated)	The data is of a time stamp in seconds. **Deprecated As Of Version:** 5.7 **Reason:** This value was accidently carried over from the win-def:EntityStateAdstypeType as it was used as a template for the ind-def:EntityStateLdaptypeType. **Comment:** This value has been deprecated and will be removed in version 6.0 of the language.

Continued on next page

Table 214 – continued from previous page

Value	Description
LDAPTYPE_EMAIL (Deprecated)	The data is of an e-mail message. **Deprecated As Of Version:** 5.7 **Reason:** This value was accidently carried over from the win-def:EntityStateAdstypeType as it was used as a template for the ind-def:EntityStateLdaptypeType. **Comment:** This value has been deprecated and will be removed in version 6.0 of the language.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateWindowsViewType ==

The EntityStateWindowsViewType restricts a string value to a specific set of values: 32-bit and 64-bit. These values describe the different values possible for the windows view behavior.

Restricts: oval-def:EntityStateStringType

Table 215: Enumeration Values

Value	Description
32_bit	Indicates the 32_bit windows view.
64_bit	Indicates the 64_bit windows view.
	The empty string value is permitted here to allow for empty elements associated with variable references.

Open Vulnerability and Assessment Language: Independent System Characteristics

- Schema: Independent System Characteristics
- Version: 5.11.1:1.2
- Release Date: 11/30/2016 09:00:00 AM

This document outlines the items of the OVAL System Characteristics XML schema that are independent of any specific family or platform. Each iten is an extention of a basic System Characteristics item defined in the core System Characteristics XML schema.

The OVAL Schema is maintained by the 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.cisecurity.org.

Item Listing

- *< family_item >*
- *< filehash_item > (Deprecated)*
- *< filehash58_item >*
- *< environmentvariable_item > (Deprecated)*
- *< environmentvariable58_item >*
- *< ldap_item >*
- *< ldap57_item > (Deprecated)*
- *< sql_item > (Deprecated)*
- *< sql57_item >*
- *< textfilecontent_item >*
- *< variable_item >*
- *< xmlfilecontent_item >*

< family_item >

This element stores high level system OS type, otherwise known as the family.

Extends: oval-sc:ItemType

Child Elements

Table 216: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
family	ind-sc:EntityItemFamilyType (0..1)	This element describes the high level system OS type, otherwise known as the family.

< filehash_item > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.8
- Reason: Replaced by the filehash58_item which allows the hash algorithm to be specified when collecting data. See the filehash58_item.
- Comment: This item has been deprecated and may be removed in a future version of the language.

This element stores the different hash values associated with a specific file.

Extends: oval-sc:ItemType

Child Elements

Table 217: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-sc:EntityItemStringType (0..1)	The filepath element specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath.
path	oval-sc:EntityItemStringType (0..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-sc:EntityItemStringType (0..1)	The name of the file.
md5	oval-sc:EntityItemStringType (0..1)	The md5 hash of the file
sha1	oval-sc:EntityItemStringType (0..1)	The sha1 hash of the file
windows_view	ind-sc:EntityItemWindowsViewType (0..1)	The windows view value from which this OVAL Item was collected. This is used to indicate from which view (32-bit or 64-bit), the associated Item was collected. A value of '32_bit' indicates the Item was collected from the 32-bit view. A value of '64-bit' indicates the Item was collected from the 64-bit view. Omitting this entity removes any assertion about which view the Item was collected from, and therefore it is strongly suggested that this entity be set. This entity only applies to 64-bit Microsoft Windows operating systems.

< filehash58_item >

This element stores a hash value associated with a specific file.

Extends: oval-sc:ItemType

Child Elements

Table 218: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-sc:EntityItemStringType (0..1)	The filepath element specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath.
path	oval-sc:EntityItemStringType (0..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-sc:EntityItemStringType (0..1)	The name of the file.
hash_type	ind-sc:EntityItemHashTypeType (0..1)	Identifier for the hash algorithm used to calculate the hash.
hash	oval-sc:EntityItemStringType (0..1)	The result of applying the hash algorithm to the file.
windows_view	ind-sc:EntityItemWindowsViewType (0..1)	The windows view value from which this OVAL Item was collected. This is used to indicate from which view (32-bit or 64-bit), the associated Item was collected. A value of '32_bit' indicates the Item was collected from the 32-bit view. A value of '64-bit' indicates the Item was collected from the 64-bit view. Omitting this entity removes any assertion about which view the Item was collected from, and therefore it is strongly suggested that this entity be set. This entity only applies to 64-bit Microsoft Windows operating systems.

< environmentvariable_item > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.8
- Reason: Replaced by the environmentvariable58_item. This item allows the hash algorithm to be specified. See the filehash58_item.
- Comment: This object has been deprecated and may be removed in a future version of the language.

This item stores information about environment variables and their values.

Extends: oval-sc:ItemType

Child Elements

Table 219: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	This element describes the name of an environment variable.
value	oval-sc:EntityItemAnySimpleType (0..1)	The actual value of the specified environment variable.

< environmentvariable58_item >

This item stores information about an environment variable, the process ID of the process from which it was retrieved, and its corresponding value.

Extends: oval-sc:ItemType

Child Elements

Table 220: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
pid	oval-sc:EntityItemIntType (0..1)	The process ID of the process from which the environment variable was retrieved.
name	oval-sc:EntityItemStringType (0..1)	This element describes the name of an environment variable.
value	oval-sc:EntityItemAnySimpleType (0..1)	The actual value of the specified environment variable.

< ldap_item >

This element holds information about specific entries in the LDAP directory. It extends the standard ItemType as defined in the oval-system-characteristics schema and one should refer to the ItemType description for more information.

Extends: oval-sc:ItemType

Child Elements

Table 221: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
suffix	oval-sc:EntityItemStringType (0..1)	Each object in an LDAP directory exists under a certain suffix (also known as a naming context). A suffix is defined as a single object in the Directory Information Tree (DIT) with every object in the tree subordinate to it.
relative_dn	oval-sc:EntityItemStringType (0..1)	The relative_dn field is used to uniquely identify an item inside the specified suffix. It contains all of the parts of the item's distinguished name except those outlined by the suffix. If the xsi:nil attribute is set to true, then the item being represented is the higher level suffix.
attribute	oval-sc:EntityItemStringType (0..1)	Specifies a named value contained by the object.
object_class	oval-sc:EntityItemStringType (0..1)	The name of the class of which the object is an instance.
ldaptype	ind-sc:EntityItemLdaptypeType (0..1)	Specifies the type of information that the specified attribute represents.
value	oval-sc:EntityItemAnySimpleType (0..unbounded)	The actual value of the specified LDAP attribute.

< ldap57_item > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.11.2
- Reason: Use the original ldap_item. The ldap57_test suffers from ambiguity; it was never adequately specified, and it does not even seem possible to have structured data in the context of the enumerated LdaptypeTypes. Use the original ldap_test instead.
- Comment: This test has been deprecated and will be removed in version 6.0 of the language.

This element holds information about specific entries in the LDAP directory. It extends the standard ItemType as defined in the oval-system-characteristics schema and one should refer to the ItemType description for more information.

Extends: oval-sc:ItemType

Child Elements

Table 222: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
suffix	oval-sc:EntityItemStringType (0..1)	Each object in an LDAP directory exists under a certain suffix (also known as a naming context). A suffix is defined as a single object in the Directory Information Tree (DIT) with every object in the tree subordinate to it.
relative_dn	oval-sc:EntityItemStringType (0..1)	The relative_dn field is used to uniquely identify an item inside the specified suffix. It contains all of the parts of the item's distinguished name except those outlined by the suffix. If the xsi:nil attribute is set to true, then the item being represented is the higher level suffix.
attribute	oval-sc:EntityItemStringType (0..1)	Specifies a named value contained by the object.
object_class	oval-sc:EntityItemStringType (0..1)	The name of the class of which the object is an instance.
ldaptype	ind-sc:EntityItemLdaptypeType (0..1)	Specifies the type of information that the specified attribute represents.
value	oval-sc:EntityItemRecordType (0..unbounded)	The actual value of the specified LDAP attribute. Note that while an LDAP attribute can contain structured data where it is necessary to collect multiple related fields that can be described by the 'record' datatype, it is not always the case. It also is possible that an LDAP attribute can contain only a single value or an array of values. In these cases, there is not a name to uniquely identify the corresponding field(s) which is a requirement for fields in the 'record' datatype. As a result, the name of the LDAP attribute will be used to uniquely identify the field(s) and satisfy this requirement. If the LDAP attribute contains a single value, the 'record' will have a single field identified by the name of the LDAP attribute. If the LDAP attribute contains an array of values, the 'record' will have multiple fields all identified by the name of the LDAP attribute.

< sql_item > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.7
- Reason: Replaced by the sql57_item. This item allows for single fields to be selected from a database. A new item was created to allow more than one field to be selected in one statement. See the sql57_item.
- Comment: This object has been deprecated and may be removed in a future version of the language.

The sql_item outlines information collected from a database via an SQL query.

Extends: oval-sc:ItemType

Child Elements

Table 223: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
engine	ind-sc:EntityItemEngineType (0..1)	The engine entity identifies the specific database engine used to connect to the database.
version	oval-sc:EntityItemStringType (0..1)	The version entity identifies the version of the database engine used to connect to the database.
connection_string	oval-sc:EntityItemStringType (0..1)	The connection_string entity defines connection parameters used to connect to the specific database.
sql	oval-sc:EntityItemStringType (0..1)	The sql entity holds the specific query used to identify the object(s) in the database.
result	oval-sc:EntityItemAnySimpleType (0..unbounded)	The result entity specifies the result(s) of the given SQL query against the database.

< sql57_item >

The sql57_item outlines information collected from a database via an SQL query.

Extends: oval-sc:ItemType

Child Elements

Table 224: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
engine	ind-sc:EntityItemEngineType (0..1)	The engine entity identifies the specific database engine used to connect to the database.
version	oval-sc:EntityItemStringType (0..1)	The version entity identifies the version of the database engine used to connect to the database.
connection_string	oval-sc:EntityItemStringType (0..1)	The connection_string entity defines connection parameters used to connect to the specific database.
sql	oval-sc:EntityItemStringType (0..1)	The sql entity holds the specific query used to identify the object(s) in the database.
result	oval-sc:EntityItemRecordType (0..unbounded)	The result entity holds the results of the specified SQL statement.

< textfilecontent_item >

The textfilecontent_item looks at the contents of a text file (aka a configuration file) by looking at individual lines.

Extends: oval-sc:ItemType

Child Elements

Table 225: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-sc:EntityItemStringType (0..1)	The filepath element specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath.
path	oval-sc:EntityItemStringType (0..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-sc:EntityItemStringType (0..1)	The filename entity specifies the name of the file (without the path) that is being represented.
pattern	oval-sc:EntityItemStringType (0..1)	The pattern entity represents a regular expression that is used to define a block of text. Subexpression notation (parenthesis) is used to call out a value(s) to test against. 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 inbetween. Note that if the pattern can match more than one block of text starting at the same point, then it matches the longest. 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.
instance	oval-sc:EntityItemIntType (0..1)	The instance entity calls out which match of the pattern is being represented by this item. The first match is given an instance value of 1, the second match is given an instance value of 2, and so on. The main purpose of this entity is too provide uniqueness for different textfilecontent_items that results from multiple matches of a given pattern against the same file.
line (Deprecated)	oval-sc:EntityItemStringType (0..1)	The line element represents a line in the file and is represented using a regular expression.
text	oval-sc:EntityItemAnySimpleType (0..1)	The text entity represents the block of text that matched the specified pattern.
subexpression	oval-sc:EntityItemAnySimpleType (0..unbounded)	The subexpression entity represents the value of a subexpression in the specified pattern. If multiple subexpressions are specified in the pattern, then multiple entities are presented. Note that the textfilecontent_state in the definition schema only allows a single subexpression entity. This means that the test will check that all (or at least one, none, etc.) the subexpressions pass the same check. This means that the order of multiple subexpression entities in the item does not matter.
windows_view	ind-sc:EntityItemWindowsViewType (0..1)	The windows view value from which this OVAL Item was collected. This is used to indicate from which view (32-bit or 64-bit), the associated Item was collected. A value of '32_bit' indicates the Item was collected from the 32-bit view. A value of '64-bit' indicates the Item was collected from the 64-bit view. Omitting this entity removes any assertion about which view the Item was collected from, and therefore it is strongly suggested that this entity be set. This entity only applies to 64-bit Microsoft Windows operating systems.

< variable_item >

This item stores information about OVAL Variables and their values.

Extends: oval-sc:ItemType

Child Elements

Table 226: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
var_ref	ind-sc:EntityItemVariableRefType (0..1)	The id of the variable.
value	oval-sc:EntityItemAnySimpleType (0..unbounded)	The value of the variable. If a variable represents and array of values, then multiple value elements should exist.

< xmlfilecontent_item >

This item stores results from checking the contents of an xml file.

Extends: oval-sc:ItemType

Child Elements

Table 227: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-sc:EntityItemStringType (0..1)	The filepath element specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath.
path	oval-sc:EntityItemStringType (0..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-sc:EntityItemStringType (0..1)	The filename element specifies the name of the file.
xpath	oval-sc:EntityItemStringType (0..1)	Specifies an XPath 1.0 expression to evaluate against the XML file specified by the filename entity. This XPath 1.0 expression must evaluate to a list of zero or more text values which will be accessible in OVAL via instances of the value_of entity. Any results from evaluating the XPath 1.0 expression other than a list of text strings (e.g., a nodes set) is considered an error. The intention is that the text values be drawn from instances of a single, uniquely named element or attribute. However, an OVAL interpreter is not required to verify this, so the author should define the XPath expression carefully. Note that “equals” is the only valid operator for the xpath entity.
value_of	oval-sc:EntityItemAnySimpleType (0..unbounded)	The value_of element checks the value(s) of the text node(s) or attribute(s) found. How this is used is entirely controlled by operator attributes.
windows_view	ind-sc:EntityItemWindowsViewType (0..1)	The windows view value from which this OVAL Item was collected. This is used to indicate from which view (32-bit or 64-bit), the associated Item was collected. A value of ‘32_bit’ indicates the Item was collected from the 32-bit view. A value of ‘64-bit’ indicates the Item was collected from the 64-bit view. Omitting this entity removes any assertion about which view the Item was collected from, and therefore it is strongly suggested that this entity be set. This entity only applies to 64-bit Microsoft Windows operating systems.

== EntityItemEngineType ==

The EntityItemEngineType complex type defines a string entity value that is restricted to an enumeration. Each valid entry in the enumeration is a valid database engine.

Restricts: oval-sc:EntityItemStringType

Table 228: Enumeration Values

Value	Description
access	The access value describes the Microsoft Access database engine.
db2	The db2 value describes the IBM DB2 database engine.
cache	The cache value describes the InterSystems Cache database engine.
firebird	The firebird value describes the Firebird database engine.
firstsql	The firstsql value describes the FirstSQL database engine.
foxpro	The foxpro value describes the Microsoft FoxPro database engine.
informix	The informix value describes the IBM Informix database engine.
ingres	The ingres value describes the Ingres database engine.
interbase	The interbase value describes the Embarcadero Technologies InterBase database engine.
lightbase	The lightbase value describes the Light Infocon LightBase database engine.
maxdb	The maxdb value describes the SAP MaxDB database engine.
monetdb	The monetdb value describes the MonetDB SQL database engine.
mimer	The mimer value describes the Mimer SQL database engine.

== EntityItemFamilyType ==

The EntityItemFamilyType complex type defines a string entity value that is restricted to a set of enumerations. Each valid enumeration is a high-level family of system operating system.

Restricts: oval-sc:EntityItemStringType

Table 229: Enumeration Values

Value	Description
android	The android value describes the Android mobile operating system.
apple_ios	The apple_ios value describes the iOS mobile operating system.
asa	The asa value describes the Cisco ASA security devices.
catos	The catos value describes the Cisco CatOS operating system.
ios	The ios value describes the Cisco IOS operating system.
iosxe	The iosxe value describes the Cisco IOS-XE operating system.
junos	The junos value describes the Juniper JunOS operating system.
macos	The macos value describes the Mac operating system.
pixos	The pixos value describes the Cisco PIX operating system.
undefined	The undefined value is to be used when the desired family is not available.
unix	The unix value describes the UNIX operating system.
vmware_infrastructure	The vmware_infrastructure value describes VMWare Infrastructure.
windows	The windows value describes the Microsoft Windows operating system.

== EntityItemHashTypeType ==

The EntityItemHashTypeType complex type restricts a string value to a specific set of values that specify the different hash algorithms that are supported. The empty string is also allowed to support empty elements associated with variable references.

Restricts: oval-sc:EntityItemStringType

Table 230: Enumeration Values

Value	Description
MD5	The MD5 hash algorithm.
SHA-1	The SHA-1 hash algorithm.
SHA-224	The SHA-224 hash algorithm.
SHA-256	The SHA-256 hash algorithm.
SHA-384	The SHA-384 hash algorithm.
SHA-512	The SHA-512 hash algorithm.
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemVariableRefType ==

The EntityItemVariableRefType complex type defines a string item entity that has a valid OVAL variable id as the value.

Restricts: oval-sc:EntityItemStringType

Pattern: oval:[**A-Za-z0-9_**-.]+:var:[1-9][0-9]*

== EntityItemLdaptypeType ==

The EntityItemLdaptypeType complex type restricts a string value to a specific set of values that specify the different types of information that an ldap attribute can represent. The empty string value is permitted here to allow for detailed error reporting.

Restricts: oval-sc:EntityItemStringType

Table 231: Enumeration Values

Value	Description
LDAPTYPE_ACI_ITEM	ACI Item, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.1
LDAPTYPE_ACCESS_POINT	Access Point, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.2
LDAPTYPE_ATTRIBUTE_TYPE_DESCRIP_STRING	Attribute Type Description, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.3
LDAPTYPE_AUDIO	Audio, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.4
LDAPTYPE_BINARY	Binary, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.5
LDAPTYPE_BIT_STRING	Bit String, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.6
LDAPTYPE_BOOLEAN	Boolean, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.7
LDAPTYPE_CERTIFICATE	Certificate, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.8
LDAPTYPE_CERTIFICATE_LIST	Certificate List, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.9
LDAPTYPE_CERTIFICATE_PAIR	Certificate Pair, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.10
LDAPTYPE_COUNTRY_STRING	Country String, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.11
LDAPTYPE_DN_STRING	DN, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.12

Continued on next page

Table 231 – continued from previous page

Value	Description
LDAPTYPE_DATA_QUALITY_SYNTAX	Data Quality Syntax, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.13
LDAPTYPE_DELIVERY_METHOD	Delivery Method, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.14
LDAPTYPE_DIRECTORY_STRING	Directory String, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.15
LDAPTYPE_DIR_CONTENT_RULE_DESCRIPTION	DIT Content Rule Description, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.16
LDAPTYPE_DIT_STRUCTURE_RULE_DESCRIPTION	DIT Structure Rule Description, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.17
LDAPTYPE_DL_SUBMIT_PERMISSION	DL Submit Permission, corresponding to OID Y 1.3.6.1.4.1.1466.115.121.1.18
LDAPTYPE_DSA_QUALITY_SYNTAX	DSA Quality Syntax, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.19
LDAPTYPE_DSE_TYPE	DSE Type, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.20
LDAPTYPE_ENHANCED_GUIDE	Enhanced Guide, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.21
LDAPTYPE_FAX_TEL_NUMBER	Facsimile Telephone Number, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.22
LDAPTYPE_FAX	Fax, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.23
LDAPTYPE_GENERALIZED_TIME	Generalized Time, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.24

Continued on next page

Table 231 – continued from previous page

Value	Description
LDAPTYPE_GUIDE	Guide, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.25
LDAPTYPE_IA5_STRING	IA5 String, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.26
LDAPTYPE_INTEGER	INTEGER, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.27
LDAPTYPE_JPEG	JPEG, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.28
LDAPTYPE_LDAP_SYNTAX_DESCRIPTION	LDAP Syntax Description, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.54
LDAPTYPE_LDAP_SCHEMA_DEFINITION	LDAP Schema Definition, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.56
LDAPTYPE_LDAP_SCHEMA_DESCRIPTION	LDAP Schema Description, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.57
LDAPTYPE_MASTER_AND_SHADOW_ACCESS_POINTS	Master And Shadow Access Points, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.29
LDAPTYPE_MATCHING_RULE_DESCRIPTION	Matching Rule Description, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.30
LDAPTYPE_MATCHING_RULE_USE_DESCRIPTION	Matching Rule Use Description, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.31
LDAPTYPE_MAIL_PREFERENCE	Mail Preference, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.32
LDAPTYPE_MHS_OR_ADDRESS	MHS OR Address, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.33

Continued on next page

Table 231 – continued from previous page

Value	Description
LDAPTYPE_MODIFY_RIGHTS	Modify Rights, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.55
LDAPTYPE_NAME_AND_OPTIONAL_UID	Name And Optional UID, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.34
LDAPTYPE_NAME_FORM_DESCRIPTION	Name Form Description, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.35
LDAPTYPE_NUMERIC_STRING	Numeric String, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.36
LDAPTYPE_OBJECT_CLASS_DESCRIP_STRING	Object Class Description, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.37
LDAPTYPE_OCTET_STRING	Octet String, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.40
LDAPTYPE_OID	OID, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.38
LDAPTYPE_MAILBOX	Other Mailbox, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.39
LDAPTYPE_POSTAL_ADDRESS	Postal Address, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.41
LDAPTYPE_PROTOCOL_INFORMATION	Protocol Information, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.42
LDAPTYPE_PRESENTATION_ADDRESS	Presentation Address, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.43
LDAPTYPE_PRINTABLE_STRING	Printable String, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.44

Continued on next page

Table 231 – continued from previous page

Value	Description
LDAPTYPE_SUBSTRING_ASSERTION	Substring Assertion, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.58
LDAPTYPE_SUBTREE_SPECIFICATION	Subtree Specification, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.45
LDAPTYPE_SUPPLIER_INFORMATION	Supplier Information, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.46
LDAPTYPE_SUPPLIER_OR_CONSUMER	Supplier Or Consumer, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.47
LDAPTYPE_SUPPLIER_AND_CONSUMER	Supplier And Consumer, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.48
LDAPTYPE_SUPPORTED_ALGORITHM	Supported Algorithm, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.49
LDAPTYPE_TELEPHONE_NUMBER	Telephone Number, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.50
LDAPTYPE_TELEX_TERMINAL_ID	Teletex Terminal Identifier, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.51
LDAPTYPE_TELEX_NUMBER	Telex Number, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.52
LDAPTYPE_UTC_TIME	UTC Time, corresponding to OID 1.3.6.1.4.1.1466.115.121.1.53
LDAPTYPE_TIMESTAMP (Deprecated)	The data is of a time stamp in seconds. **Deprecated As Of Version:** 5.7 **Reason:** This value was accidently carried over from the win-sc:EntityItemAdstypeType as it was used as a template for the ind-sc:EntityItemLdaptypeType. **Comment:** This value has been deprecated and will be removed in version 6.0 of the language.

Continued on next page

Table 231 – continued from previous page

Value	Description
LDAPTYPE_EMAIL (Deprecated)	The data is of an e-mail message. **Deprecated As Of Version:** 5.7 **Reason:** This value was accidently carried over from the win-sc:EntityItemAdstypeType as it was used as a template for the ind-sc:EntityItemLdaptypeType. **Comment:** This value has been deprecated and will be removed in version 6.0 of the language.
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemWindowsViewType ==

The EntityItemWindowsViewType restricts a string value to a specific set of values: 32-bit and 64-bit. These values describe the different values possible for the windows view behavior.

Restricts: oval-sc:EntityItemStringType

Table 232: Enumeration Values

Value	Description
32_bit	Indicates the 32_bit windows view.
64_bit	Indicates the 64_bit windows view.
	The empty string value is permitted here to allow for empty elements associated with variable references.

Open Vulnerability and Assessment Language: Apple iOS Definition

- Schema: Apple iOS Definition
- Version: 5.11.1:1.2
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the Apple iOS specific tests found in Open Vulnerability and Assessment Language (OVAL). Each item is an extension of the standard item element defined in the Core Definition Schema. Through extension, each item inherits a set of elements and attributes that are shared amongst all OVAL Items. Each item is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core Definition Schema is not outlined here.

The OVAL Schema is maintained by the 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.cisecurity.org.

See public documentation at https://developer.apple.com/library/ios/featuredarticles/iPhoneConfigurationProfileRef/Introduction/Introduction.html

Test Listing

- *< globalrestrictions_test >*
- *< passcodepolicy_test >*
- *< profile_test >*

< globalrestrictions_test >

The globalrestrictions_test is used to check the status of the global restrictions in place on the device. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a globalrestrictions_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 233: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< globalrestrictions_object >

The globalrestrictions_object element is used by a global restrictions test to define those objects to be evaluated based on a specified state. Any OVAL Test written to check global restrictions status will reference the same globalrestrictions_object which is basically an empty object element.

Extends: oval-def:ObjectType

< globalrestrictions_state >

Information on global restrictions in place on the device

Extends: oval-def:StateType

Child Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
allow_account_modification	oval-def:EntityStateBoolType (0..1)	Optional. Supervised only. If set to false,
allow_airdrop	oval-def:EntityStateBoolType (0..1)	Optional. Supervised only. If set to false,
allow_app_cellular_data_modification	oval-def:EntityStateBoolType (0..1)	Optional. Supervised only. If set to false,
allow_app_installation	oval-def:EntityStateBoolType (0..1)	Optional. When false, the App Store is d
allow_assistant	oval-def:EntityStateBoolType (0..1)	Optional. When false, disables Siri. Defa
allow_assistant_user_generated_content	oval-def:EntityStateBoolType (0..1)	Optional. Supervised only. When false, p
allow_assistant_while_locked	oval-def:EntityStateBoolType (0..1)	Optional. When false, the user is unable
allow_bookstore	oval-def:EntityStateBoolType (0..1)	Optional. Supervised only. If set to false,
allow_bookstore_erotica	oval-def:EntityStateBoolType (0..1)	Optional. Supervised only prior to iOS 6
allow_camera	oval-def:EntityStateBoolType (0..1)	Optional. When false, the camera is com
allow_cloud_backup	oval-def:EntityStateBoolType (0..1)	Optional. When false, disables backing u
allow_cloud_document_sync	oval-def:EntityStateBoolType (0..1)	Optional. When false, disables document
allow_cloud_keychain_sync	oval-def:EntityStateBoolType (0..1)	Optional. If false, disables keychain sync
allow_diagnostic_submission	oval-def:EntityStateBoolType (0..1)	Optional. When false, this prevents the d
allow_explicit_content	oval-def:EntityStateBoolType (0..1)	Optional. When false, explicit music or v
allow_find_my_friends_modification	oval-def:EntityStateBoolType (0..1)	Optional. Supervised only. If set to false,
allow_fingerprint_for_unlock	oval-def:EntityStateBoolType (0..1)	Optional. If false, prevents Touch ID fro
allow_game_center	oval-def:EntityStateBoolType (0..1)	Optional. Supervised only. When false, G
allow_host_pairing	oval-def:EntityStateBoolType (0..1)	Supervised only. If set to false, host pairi
allow_lock_screen_control_center	oval-def:EntityStateBoolType (0..1)	Optional. If false, prevents Control Cente
allow_lock_screen_notifications_view	oval-def:EntityStateBoolType (0..1)	Optional. If set to false, the Notifications
allow_lock_screen_today_view	oval-def:EntityStateBoolType (0..1)	Optional. If set to false, the Today view i
allow_open_from_managed_to_unmanaged	oval-def:EntityStateBoolType (0..1)	Optional. If false, documents in managed
allow_open_from_unmanaged_to_managed	oval-def:EntityStateBoolType (0..1)	Optional. If set to false, documents in un
allow_ota_pki_updates	oval-def:EntityStateBoolType (0..1)	Optional. If false, over-the-air PKI updat
allow_passbook_while_locked	oval-def:EntityStateBoolType (0..1)	Optional. If set to false, Passbook notific
allow_photo_stream	oval-def:EntityStateBoolType (0..1)	Optional. When false, disables Photo Str
allow_safari	oval-def:EntityStateBoolType (0..1)	Optional. When false, the Safari web bro
allow_screen_shot	oval-def:EntityStateBoolType (0..1)	Optional. When false, users are unable to
allow_shared_stream	oval-def:EntityStateBoolType (0..1)	Optional. If set to false, Shared Photo Str
allow_ui_configuration_profile_installation	oval-def:EntityStateBoolType (0..1)	Optional. Supervised only. If set to false,
allow_untrusted_tls_prompt	oval-def:EntityStateBoolType (0..1)	Optional. When false, automatically reje
allow_voice_dialing	oval-def:EntityStateBoolType (0..1)	Optional. When false, disables voice dial
allow_youtube	oval-def:EntityStateBoolType (0..1)	Optional. When false, the YouTube appli
allow_itunes	oval-def:EntityStateBoolType (0..1)	Optional. When false, the iTunes Music S
autonomous_single_app_mode_permitted_appids	oval-def:EntityStateStringType (0..1)	Optional. If present, allows the identified
force_encrypted_backup	oval-def:EntityStateBoolType (0..1)	Optional. When true, encrypts all backup
force_itunes_store_password_entry	oval-def:EntityStateBoolType (0..1)	Optional. When true, forces user to enter
force_limit_ad_tracking	oval-def:EntityStateBoolType (0..1)	Optional. If true, limits ad tracking. Defa
safari_allow_auto_fill	oval-def:EntityStateBoolType (0..1)	Optional. When false, Safari auto-fill is d

< passcodepolicy_test >

The passcodepolicy_test is used to check the status of the passcode policy in place on the device. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a passcodepolicy_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 235: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< passcodepolicy_object >

The passcodepolicy_object element is used by a passcode policy test to define those objects to be evaluated based on a specified state. Any OVAL Test written to check passcode policy status will reference the same passcodepolicy_object which is basically an empty object element.

Extends: oval-def:ObjectType

< passcodepolicy_state >

Passcode Policy Items from public Apple Configuration Profile Reference

Extends: oval-def:StateType

Child Elements

Table 236: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
allow_simple	oval-def:EntityStateBoolType (0..1)	Optional. Default true. Determines whether a simple passcode is allowed. A simple passcode is defined as containing repeated characters, or increasing/decreasing characters (such as 123 or CBA). Setting this value to false is synonymous to setting minComplexChars to “1”.
force_pin	oval-def:EntityStateBoolType (0..1)	Optional. Default false. Determines whether the user is forced to set a PIN. Simply setting this value (and not others) forces the user to enter a passcode, without imposing a length or quality.
max_failed_attempts	oval-def:EntityStateIntType (0..1)	Optional. Default 11. Allowed range [2…11]. Specifies the number of allowed failed attempts to enter the passcode at the device’s lock screen. Once this number is exceeded, the device is locked and must be connected to its designated iTunes in order to be unlocked.
max_inactivity	oval-def:EntityStateIntType (0..1)	Optional. Default Infinity. Specifies the number of minutes for which the device can be idle (without being unlocked by the user) before it gets locked by the system. Once this limit is reached, the device is locked and the passcode must be entered. In OS X, this will be translated to screensaver settings.
max_pin_age_in_days	oval-def:EntityStateIntType (0..1)	Optional. Default Infinity. Specifies the number of days for which the passcode can remain unchanged. After this number of days, the user is forced to change the passcode before the device is unlocked.
min_complex_chars	oval-def:EntityStateIntType (0..1)	Optional. Default 0. Specifies the minimum number of complex characters that a passcode must contain. A “complex” character is a character other than a number or a letter.
min_length	oval-def:EntityStateIntType (0..1)	Optional. Default 0. Specifies the minimum overall length of the passcode. This parameter is independent of the also optional minComplexChars argument.
require_alphanumeric	oval-def:EntityStateBoolType (0..1)	Optional. Default false. Specifies whether the user must enter alphabetic characters (“abcd”) or if numbers are sufficient.
pin_history	oval-def:EntityStateIntType (0..1)	Optional. When the user changes the passcode, it has to be unique within the last N entries in the history. Minimum value is 1, maximum value is 50.
max_grace_period	oval-def:EntityStateIntType (0..1)	Optional. The maximum grace period, in minutes, to unlock the phone without entering a passcode. Default is 0, that is no grace period, which requires a passcode immediately. In OS X, this will be translated to screensaver settings.

< profile_test >

The profile_test is used to check the status of the profiles in place on the device. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a profile_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 237: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< profile_object >

The profile_object element is used by a profile test to define those objects to be evaluated based on a specified state. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic.

Extends: oval-def:ObjectType

Child Elements

Table 238: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
identifier	oval-def:EntityObjectStringType (1..1)	A reverse-DNS style identifier (com.example.myprofile, for example) that identifies the profile. This string is used to determine whether a new profile should replace an existing one or should be added.
uuid	oval-def:EntityObjectStringType (1..1)	A globally unique identifier for the payload. The actual content is unimportant, but it must be globally unique.
oval-def:filter	n/a (0..unbounded)	

< profile_state >

Represents information about each configuration profile installed on the device.

Extends: oval-def:StateType

Child Elements

Table 239: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
has_removal_passcode	oval-def:EntityStateBoolType (0..1)	Optional. Set to true if there is a removal passcode.
is_encrypted	oval-def:EntityStateBoolType (0..1)	Optional. Set to true if the profile is encrypted.
payload	oval-def:EntityStateRecordType (0..1)	Optional. Contains information about each payload inside the configuration profile.
description	oval-def:EntityStateStringType (0..1)	Optional. A description of the profile, shown on the Detail screen for the profile.
display_name	oval-def:EntityStateStringType (0..1)	Optional. A human-readable name for the profile. This value is displayed on the Detail screen. It does not have to be unique.
identifier	oval-def:EntityStateStringType (0..1)	A reverse-DNS style identifier (com.example.myprofile, for example) that identifies the profile. This string is used to determine whether a new profile should replace an existing one or should be added.
organization	oval-def:EntityStateStringType (0..1)	Optional. A human-readable string containing the name of the organization that provided the profile.
removal_disallowed	oval-def:EntityStateBoolType (0..1)	Optional. If present and set to true, the user cannot delete the profile (unless the profile has a removal password and the user provides it).
uuid	oval-def:EntityStateStringType (0..1)	A globally unique identifier for the payload. The actual content is unimportant, but it must be globally unique.
version	oval-def:EntityStateIntType (0..1)	The version number of the profile format. This describes the version of the configuration profile as a whole, not of the individual profiles within it. Currently, this value should be 1.

Open Vulnerability and Assessment Language: Apple iOS System Characteristics

- Schema: Apple iOS System Characteristics
- Version: 5.11.1:1.2
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the Apple iOS specific system characteristic items found in Open Vulnerability and Assessment Language (OVAL). Each item is an extension of the standard item element defined in the Core System Characteristic Schema. Through extension, each item inherits a set of elements and attributes that are shared amongst all OVAL Items. Each item is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core System Characteristic Schema is not outlined here.

The OVAL Schema is maintained by the 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.cisecurity.org.

See public documentation at https://developer.apple.com/library/ios/featuredarticles/iPhoneConfigurationProfileRef/Introduction/Introduction.html

Item Listing

- *< globalrestrictions_item >*
- *< passcodepolicy_item >*
- *< profile_item >*

< globalrestrictions_item >

Information on global restrictions in place on the device derived from Apple's public Configuration Profile reference documentation

Extends: oval-sc:ItemType

Child Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
allow_account_modification	oval-sc:EntityItemBoolType (0..1)	Optional. Supervised only. If set
allow_airdrop	oval-sc:EntityItemBoolType (0..1)	Optional. Supervised only. If set
allow_app_cellular_data_modification	oval-sc:EntityItemBoolType (0..1)	Optional. Supervised only. If set
allow_app_installation	oval-sc:EntityItemBoolType (0..1)	Optional. When false, the App S
allow_assistant	oval-sc:EntityItemBoolType (0..1)	Optional. When false, disables S
allow_assistant_user_generated_content	oval-sc:EntityItemBoolType (0..1)	Optional. Supervised only. When
allow_assistant_while_locked	oval-sc:EntityItemBoolType (0..1)	Optional. When false, the user is
allow_bookstore	oval-sc:EntityItemBoolType (0..1)	Optional. Supervised only. If set
allow_bookstore_erotica	oval-sc:EntityItemBoolType (0..1)	Optional. Supervised only prior
allow_camera	oval-sc:EntityItemBoolType (0..1)	Optional. When false, the camer
allow_cloud_backup	oval-sc:EntityItemBoolType (0..1)	Optional. When false, disables b
allow_cloud_document_sync	oval-sc:EntityItemBoolType (0..1)	Optional. When false, disables d
allow_cloud_keychain_sync	oval-sc:EntityItemBoolType (0..1)	Optional. If false, disables keych
allow_diagnostic_submission	oval-sc:EntityItemBoolType (0..1)	Optional. When false, this preve
allow_explicit_content	oval-sc:EntityItemBoolType (0..1)	Optional. When false, explicit m
allow_find_my_friends_modification	oval-sc:EntityItemBoolType (0..1)	Optional. Supervised only. If set
allow_fingerprint_for_unlock	oval-sc:EntityItemBoolType (0..1)	Optional. If false, prevents Touch
allow_game_center	oval-sc:EntityItemBoolType (0..1)	Optional. Supervised only. When
allow_host_pairing	oval-sc:EntityItemBoolType (0..1)	Supervised only. If set to false, h
allow_lock_screen_control_center	oval-sc:EntityItemBoolType (0..1)	Optional. If false, prevents Contr
allow_lock_screen_notifications_view	oval-sc:EntityItemBoolType (0..1)	Optional. If set to false, the Noti
allow_lock_screen_today_view	oval-sc:EntityItemBoolType (0..1)	Optional. If set to false, the Toda
allow_open_from_managed_to_unmanaged	oval-sc:EntityItemBoolType (0..1)	Optional. If false, documents in
allow_open_from_unmanaged_to_managed	oval-sc:EntityItemBoolType (0..1)	Optional. If set to false, documen

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
allow_ota_pki_updates	oval-sc:EntityItemBoolType (0..1)	Optional. If false, over-the-air PI
allow_passbook_while_locked	oval-sc:EntityItemBoolType (0..1)	Optional. If set to false, Passboo
allow_photo_stream	oval-sc:EntityItemBoolType (0..1)	Optional. When false, disables P
allow_safari	oval-sc:EntityItemBoolType (0..1)	Optional. When false, the Safari
allow_screen_shot	oval-sc:EntityItemBoolType (0..1)	Optional. When false, users are u
allow_shared_stream	oval-sc:EntityItemBoolType (0..1)	Optional. If set to false, Shared P
allow_ui_configuration_profile_installation	oval-sc:EntityItemBoolType (0..1)	Optional. Supervised only. If set
allow_untrusted_tls_prompt	oval-sc:EntityItemBoolType (0..1)	Optional. When false, automatic
allow_voice_dialing	oval-sc:EntityItemBoolType (0..1)	Optional. When false, disables v
allow_youtube	oval-sc:EntityItemBoolType (0..1)	Optional. When false, the YouTu
allow_itunes	oval-sc:EntityItemBoolType (0..1)	Optional. When false, the iTunes
autonomous_single_app_mode_permitted_appids	oval-sc:EntityItemStringType (0..unbounded)	Optional. If present, allows the id
force_encrypted_backup	oval-sc:EntityItemBoolType (0..1)	Optional. When true, encrypts al
force_itunes_store_password_entry	oval-sc:EntityItemBoolType (0..1)	Optional. When true, forces user
force_limit_ad_tracking	oval-sc:EntityItemBoolType (0..1)	Optional. If true, limits ad tracki
safari_allow_auto_fill	oval-sc:EntityItemBoolType (0..1)	Optional. When false, Safari auto

< passcodepolicy_item >

Passcode Policy Items from public Apple Configuration Profile Reference

Extends: oval-sc:ItemType

Child Elements

Table 241: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
allow_simple	oval-sc:EntityItemBoolType (0..1)	Optional. Default true. Determines whether a simple passcode is allowed. A simple passcode is defined as containing repeated characters, or increasing/decreasing characters (such as 123 or CBA). Setting this value to false is synonymous to setting minComplexChars to “1”.
force_pin	oval-sc:EntityItemBoolType (0..1)	Optional. Default false. Determines whether the user is forced to set a PIN. Simply setting this value (and not others) forces the user to enter a passcode, without imposing a length or quality.
max_failed_attempts	oval-sc:EntityItemIntType (0..1)	Optional. Default 11. Allowed range [2…11]. Specifies the number of allowed failed attempts to enter the passcode at the device’s lock screen. Once this number is exceeded, the device is locked and must be connected to its designated iTunes in order to be unlocked.
max_inactivity	oval-sc:EntityItemIntType (0..1)	Optional. Default Infinity. Specifies the number of minutes for which the device can be idle (without being unlocked by the user) before it gets locked by the system. Once this limit is reached, the device is locked and the passcode must be entered. In OS X, this will be translated to screensaver settings.
max_pin_age_in_days	oval-sc:EntityItemIntType (0..1)	Optional. Default Infinity. Specifies the number of days for which the passcode can remain unchanged. After this number of days, the user is forced to change the passcode before the device is unlocked.
min_complex_chars	oval-sc:EntityItemIntType (0..1)	Optional. Default 0. Specifies the minimum number of complex characters that a passcode must contain. A “complex” character is a character other than a number or a letter.
min_length	oval-sc:EntityItemIntType (0..1)	Optional. Default 0. Specifies the minimum overall length of the passcode. This parameter is independent of the also optional minComplexChars argument.
require_alphanumeric	oval-sc:EntityItemBoolType (0..1)	Optional. Default false. Specifies whether the user must enter alphabetic characters (“abcd”), or if numbers are sufficient.
pin_history	oval-sc:EntityItemIntType (0..1)	Optional. When the user changes the passcode, it has to be unique within the last N entries in the history. Minimum value is 1, maximum value is 50.
max_grace_period	oval-sc:EntityItemIntType (0..1)	Optional. The maximum grace period, in minutes, to unlock the phone without entering a passcode. Default is 0, that is no grace period, which requires a passcode immediately. In OS X, this will be translated to screensaver settings.

< profile_item >

Represents information about each configuration profile installed on the device.

Extends: oval-sc:ItemType

Child Elements

Table 242: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
has_removal_passcode	oval-sc:EntityItemBoolType (0..1)	Optional. Set to true if there is a removal passcode.
is_encrypted	oval-sc:EntityItemBoolType (0..1)	Optional. Set to true if the profile is encrypted.
payload	oval-sc:EntityItemRecordType (0..unbounded)	Optional. Contains information about each payload inside the configuration profile.
description	oval-sc:EntityItemStringType (0..1)	Optional. A description of the profile, shown on the Detail screen for the profile.
display_name	oval-sc:EntityItemStringType (0..1)	Optional. A human-readable name for the profile. This value is displayed on the Detail screen. It does not have to be unique.
identifier	oval-sc:EntityItemStringType (0..1)	A reverse-DNS style identifier (com.example.myprofile, for example) that identifies the profile. This string is used to determine whether a new profile should replace an existing one or should be added.
organization	oval-sc:EntityItemStringType (0..1)	Optional. A human-readable string containing the name of the organization that provided the profile.
removal_disallowed	oval-sc:EntityItemBoolType (0..1)	Optional. If present and set to true, the user cannot delete the profile (unless the profile has a removal password and the user provides it).
uuid	oval-sc:EntityItemStringType (0..1)	A globally unique identifier for the payload. The actual content is unimportant, but it must be globally unique.
version	oval-sc:EntityItemIntType (0..1)	The version number of the profile format. This describes the version of the configuration profile as a whole, not of the individual profiles within it. Currently, this value should be 1.

Open Vulnerability and Assessment Language: Android Definition

- Schema: Android Definition
- Version: 5.11.1:1.1
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the Android specific tests found in Open Vulnerability and Assessment Language (OVAL). Each test is an extension of the standard test element defined in the Core Definition Schema. Through extension, each test inherits a set of elements and attributes that are shared amongst all OVAL tests. Each test is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core Definition Schema is not outlined here.

The OVAL Schema is maintained by the 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.cisecurity.org.

Test Listing

- *< appmanager_test >*
- *< bluetooth_test >*
- *< camera_test >*
- *< certificate_test >*
- *< devicesettings_test >*
- *< encryption_test >*
- *< locationservice_test >*
- *< network_test >*
- *< password_test >*
- *< systemdetails_test >*
- *< wifi_test >*
- *< wifinetwork_test >*
- *< telephony_test >*

< appmanager_test >

The appmanager_test is used to verify the applications installed on the device. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a appmanager_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 243: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< appmanager_object >

The appmanager_object element is used by a appmanager_test to define the required application properties to verify. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information.

Extends: oval-def:ObjectType

Child Elements

Table 244: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
package_name	oval-def:EntityObjectStringType (1..1)	Name of the package.
signing_certificate	oval-def:EntityObjectBinaryType (1..1)	Hexadecimal string of the signing certificate corresponding with the key used to sign the application package. Only the actual signing certificate should be included, not CA certificates in the chain (if applicable).
oval-def:filter	n/a (0..unbounded)	

< appmanager_state >

The appmanager_state element defines the application settings.

Extends: oval-def:StateType

Child Elements

Table 245: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
application_name	oval-def:EntityStateStringType (0..1)	Name of the application.
uid	oval-def:EntityStateStringType (0..1)	Linux userid assigned to the application. (In some cases multiple applications can share a userid.)
gid	oval-def:EntityStateStringType (0..unbounded)	One element for each group id that the application belongs to.
package_name	oval-def:EntityStateStringType (0..1)	Name of the package.
data_directory	oval-def:EntityStateStringType (0..1)	Data directory assigned to the application.
version	oval-def:EntityStateStringType (0..1)	Application version.
current_status	oval-def:EntityStateBoolType (0..1)	True if the application is enabled.
permission	oval-def:EntityStateStringType (0..1)	One element for each permission granted to the application.
native_lib_dir	oval-def:EntityStateStringType (0..1)	Directory where the application's native libraries (if any) have been installed.
signing_certificate	oval-def:EntityStateBinaryType (0..unbounded)	Hexadecimal string of the signing certificate corresponding with the key used to sign the application package. Only the actual signing certificate should be included, not CA certificates in the chain (if applicable).
first_install_time	oval-def:EntityStateIntType (0..1)	Time at which the app was first installed, expressed in milliseconds since January 1, 1970 00:00:00 UTC.
last_update_time	oval-def:EntityStateIntType (0..1)	Time at which the app was last updated, expressed in milliseconds since January 1, 1970 00:00:00 UTC.
package_file_location	oval-def:EntityStateStringType (0..1)	From ApplicationInfo.sourceDir, the full path to the location of the publicly available parts of the application package.

< bluetooth_test >

The bluetooth_test is used to check the status of bluetooth settings on the device. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a bluetooth_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 246: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< bluetooth_object >

The bluetooth_object element is used by a bluetooth test to define those objects to be evaluated based on a specified state. Any OVAL Test written to check bluetooth settings status will reference the same bluetooth_object which is basically an empty object element.

Extends: oval-def:ObjectType

< bluetooth_state >

The bluetooth_state element defines the bluetooth general settings status.

Extends: oval-def:StateType

Child Elements

Table 247: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
discoverable	oval-def:EntityStateBoolType (0..1)	True if device Bluetooth is currently in discoverable mode.
current_status	oval-def:EntityStateBoolType (0..1)	True if device Bluetooth is currently enabled.

< camera_test >

The camera_test is used to check camera-related information.

Extends: oval-def:TestType

Child Elements

Table 248: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< camera_object >

The camera_object element is used by a camera test to define those objects to evaluate based on a camera state.

Extends: oval-def:ObjectType

< camera_state >

The camera_state element contains a single entity that is used to check the status of the camera.

Extends: oval-def:StateType

Child Elements

Table 249: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
camera_disabled_policy	oval-def:EntityStateBoolType (0..1)	If true, then a policy is being enforced disabling use of the camera. The policy is only available in Android 4.0 and up (and potentially on older Android devices if specifically added by the device vendor).

< certificate_test >

The certificate_test is used to check the certificates installed on the device.

Extends: oval-def:TestType

Child Elements

Table 250: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< certificate_object >

The certificate_object element is used by a certificate test to define those objects to evaluate based on a certificate state.

Extends: oval-def:ObjectType

< certificate_state >

The certificate_state element contains a single entity that is used to check the status of the certificates.

Extends: oval-def:StateType

Child Elements

Table 251: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
trusted_certificate	oval-def:EntityStateBinaryType (0..unbounded)	Hexadecimal string of each certificate in the OS's trusted certificate store, including both certificates installed by the system and by users. System trusted certificates that were disabled by the user are not included here.

< devicesettings_test >

The devicesettings_test is used to check the status of various settings on the device. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a devicesettings_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 252: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< devicesettings_object >

The devicesettings_object element is used by a device settings test to define those objects to be evaluated based on a specified state. Any OVAL Test written to check device settings will reference the same devicesettings_object which is basically an empty object element.

Extends: oval-def:ObjectType

< devicesettings_state >

The devicesettings_state element defines the device settings.

Extends: oval-def:StateType

Child Elements

Table 253: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
adb_enabled	oval-def:EntityStateBoolType (0..1)	True if Android Debug Bridge (USB debugging) is enabled.
allow_mock_location	oval-def:EntityStateBoolType (0..1)	True if mock locations and location provider status can be injected into Android's Location Manager.
install_non_market_apps	oval-def:EntityStateBoolType (0..1)	True if applications can be installed from "unknown sources".
device_admin	oval-def:EntityStateStringType (0..unbounded)	One element per application that holds device administrator access. Contains the application's package name.
auto_time	oval-def:EntityStateBoolType (0..1)	True if the user prefers the date and time to be automatically fetched from the network.
auto_time_zone	oval-def:EntityStateBoolType (0..1)	True if the user prefers the time zone to be automatically fetched from the network.
usb_mass_storage_enabled	oval-def:EntityStateBoolType (0..1)	True if USB mass storage is enabled on the device, otherwise false.

< encryption_test >

The encryption_test is used to check the encryption status on the device. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a encryption_object and the optional state element references a encryption_state that specifies the information to check.

Extends: oval-def:TestType

Child Elements

Table 254: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< encryption_object >

The encryption_object element is used by a encryption test to define those objects to evaluated based on a specified state. Any OVAL Test written to check encryption settings will reference the same encryption_object which is basically an empty object element.

Extends: oval-def:ObjectType

< encryption_state >

The encryption_state element defines the encryption settings configured on the device.

Extends: oval-def:StateType

Child Elements

Table 255: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
encryption_policy_enabled	oval-def:EntityStateBoolType (0..1)	True if a policy is in place requiring the device storage to be encrypted. (android.app.admin.DevicePolicyManager.getStorageEncryption())
encryption_status	android-def:EntityStateEncryptionStatusType (0..1)	The current status of device encryption. (android.app.admin.DevicePolicyManager.getStorageEncryptionStatus()) Either ENCRYPTION_STATUS_UNSUPPORTED, ENCRYPTION_STATUS_INACTIVE, ENCRYPTION_STATUS_ACTIVATING, or ENCRYPTION_STATUS_ACTIVE as documented in the Android SDK's DevicePolicyManager class.

< locationservice_test >

The locationservice_test is used to check the status of location based services. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a locationservice_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 256: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< locationservice_object >

The locationservice_object element is used by a location service test to define those objects to evaluated based on a specified state. Any OVAL Test written to check location based services status will reference the same locationservice_object which is basically an empty object element.

Extends: oval-def:ObjectType

< locationservice_state >

The locationservice_state element defines the location based services status.

Extends: oval-def:StateType

Child Elements

Table 257: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
gps_enabled	oval-def:EntityStateBoolType (0..1)	A boolean value indicating whether the GPS location provider is enabled.
network_enabled	oval-def:EntityStateBoolType (0..1)	A boolean value indicating whether the network location provider is enabled.

< network_test >

The network_test is used to check the status of network preferences on the device. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a network_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 258: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< network_object >

The network_object element is used by a network test to define those objects to be evaluated based on a specified state. Any OVAL Test written to check network preference will reference the same network_object which is basically an empty object element.

Extends: oval-def:ObjectType

< network_state >

The network_state element defines the network preferences.

Extends: oval-def:StateType

Child Elements

Table 259: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
airplane_mode	oval-def:EntityStateBoolType (0..1)	True if airplane mode is enabled on the device.
nfc_enabled	oval-def:EntityStateBoolType (0..1)	True if NFC is enabled on the device.

< password_test >

The password test is used to check specific policy associated with passwords and the device screen lock. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a password_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 260: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< password_object >

The password_object element is used by a password test to define those objects to evaluated based on a specified state. Any OVAL Test written to check password policy will reference the same password_object which is basically an empty object element.

Extends: oval-def:ObjectType

< password_state >

The password_state element specifies the various policies associated with passwords and the device screen lock. A password test will reference a specific instance of this state that defines the exact settings that need to be evaluated.

Extends: oval-def:StateType

Child Elements

Table 261: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
max_num_failed_user_auth	oval-def:EntityStateIntType (0..1)	Maximum number of failed user authentications before device wipe. Zero means there is no policy in place.
password_hist_len	oval-def:EntityStateIntType (0..1)	Specifies the length of password history maintained (passwords in the history cannot be reused). Zero means there is no policy in place.
password_quality	android-def:EntityStatePasswordQualityType (0..1)	The current minimum required password quality required by device policy. Represented as a string corresponding with a valid Android password quality, currently one of: PASSWORD_QUALITY_ALPHABETIC PASSWORD_QUALITY_ALPHANUMERIC PASSWORD_QUALITY_BIOMETRIC_WEAK PASSWORD_QUALITY_COMPLEX PASSWORD_QUALITY_NUMERIC PASSWORD_QUALITY_SOMETHING PASSWORD_QUALITY_UNSPECIFIED
password_min_length	oval-def:EntityStateIntType (0..1)	Minimum length of characters password must have. This constraint is only imposed if the password quality is one of PASSWORD_QUALITY_NUMERIC, PASSWORD_QUALITY_ALPHABETIC, PASSWORD_QUALITY_ALPHANUMERIC, or PASSWORD_QUALITY_COMPLEX.
password_min_letters	oval-def:EntityStateIntType (0..1)	Minimum number of letters password must have. This constraint is only imposed if the password quality is PASSWORD_QUALITY_COMPLEX.
password_min_lowercase_letters	oval-def:EntityStateIntType (0..1)	Minimum number of lower case letters password must have. This constraint is only imposed if the password quality is PASSWORD_QUALITY_COMPLEX.
password_min_nonletters	oval-def:EntityStateIntType (0..1)	Minimum number of non-letter characters password must have. This constraint is only imposed if the password quality is PASSWORD_QUALITY_COMPLEX.
password_min_numeric	oval-def:EntityStateIntType (0..1)	Minimum number of numeric characters password must have. This constraint is only imposed if the password quality is PASSWORD_QUALITY_COMPLEX.
password_min_symbols	oval-def:EntityStateIntType (0..1)	Minimum number of symbol characters password must have. This constraint is only imposed if the password quality is PASSWORD_QUALITY_COMPLEX.
password_min_uppercase_letters	oval-def:EntityStateIntType (0..1)	Minimum number of upper case letters password must have. This constraint is only imposed if the password quality is PASSWORD_QUALITY_COMPLEX.
password_expiration_timeout	oval-def:EntityStateIntType (0..1)	Gets the current password expiration timeout policy, in milliseconds. Zero means there is no policy in place.
password_visible	oval-def:EntityStateBoolType (0..1)	When true, the most recently keyed in password character is shown to the user on the screen (previously entered characters are masked out). When false, all keyed in password characters are immediately masked out. This setting is manageable by the device user through the device settings.
active_password_sufficient	oval-def:EntityStateBoolType (0..1)	When true, the current device password is compliant with the password policy. (If the policy was recently established, it is possible that a password compliant with the policy may not yet be in place.)
current_failed_password_attempts	oval-def:EntityStateIntType (0..1)	The number of times the user has failed at entering a password since the last successful password entry.
screen_lock_timeout	oval-def:EntityStateIntType (0..1)	The current policy for the highest screen lock timeout the user is allowed to specify. 0 indicates no restriction. (The user may still specify lower values in the device settings.)
keyguard_disabled_features	android-def:EntityStateKeyguardDisabledFeaturesType (0..1)	The current policy for lockscreen widgets as retrieved by DevicePolicyManager.getKeyguardDisabledFeatures. May be

< systemdetails_test >

The syste_details test is used to get system hardware and operating system information. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a systemdetails_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 262: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< systemdetails_object >

The systemdetails_object element is used by a systemdetails test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information.

Extends: oval-def:ObjectType

< systemdetails_state >

The systemdetails_state element defines the information about the hardware and the operating system. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 263: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
hardware	oval-def:EntityStateStringType (0..1)	The hardware model, as provided by android.os.Build.HARDWARE using the Android SDK.
manufacturer	oval-def:EntityStateStringType (0..1)	The device manufacturer, as provided by android.os.Build.MANUFACTURER using the Android SDK.
model	oval-def:EntityStateStringType (0..1)	The device model identifier, as provided by android.os.Build.MODEL using the Android SDK.
product	oval-def:EntityStateStringType (0..1)	The product name, as provided by android.os.Build.PRODUCT using the Android SDK.
cpu_abi	oval-def:EntityStateStringType (0..1)	The name of the instruction set of native code, as provided by android.os.Build.CPU_ABI using the Android SDK.
cpu_abi2	oval-def:EntityStateStringType (0..1)	The name of the second instruction set of native code, as provided by android.os.Build.CPU_ABI2 using the Android SDK.
build_fingerprint	oval-def:EntityStateStringType (0..1)	Build fingerprint, as provided by android.os.Build.FINGERPRINT using the Android SDK.
os_version_code_name	oval-def:EntityStateStringType (0..1)	Operating system version code, as provided by android.os.Build.VERSION.CODENAME using the Android SDK.
os_version_build_number	oval-def:EntityStateStringType (0..1)	Operating system build number, as provided by android.os.Build.VERSION.INCREMENTAL using the Android SDK.
os_version_release_name	oval-def:EntityStateStringType (0..1)	Operating system release name, as provided by android.os.Build.VERSION.RELEASE using the Android SDK.
os_version_sdk_number	oval-def:EntityStateIntType (0..1)	Operating system SDK number, as provided by android.os.Build.VERSION.SDK_INT using the Android SDK.
hardware_keystore	oval-def:EntityStateBoolType (0..1)	True if the device provides a hardware backed cryptographic keystore (a hardware keystore prevents exporting private keys or directly exposing private keys to the OS), otherwise false.

< wifi_test >

The wifi_test is used to check the status of general Wi-Fi settings on the device. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a wifi_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 264: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< wifi_object >

The wifi_object element is used by a wifi test to define those objects to evaluated based on a specified state. Any OVAL Test written to check wifi settings status will reference the same wifi_object which is basically an empty object element.

Extends: oval-def:ObjectType

< wifi_state >

The wifi_state element defines the wifi general settings status.

Extends: oval-def:StateType

Child Elements

Table 265: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
wifi_status	oval-def:EntityStateBoolType (0..1)	True if Wi-Fi is currently enabled on the device.
network_availability_notification	oval-def:EntityStateBoolType (0..1)	True if the Wi-Fi network availability notification setting is currently enabled on the device.

< wifinetwork_test >

The wifinetwork_test is used to check information about the configured Wi-Fi networks on the device. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a wifinetwork_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 266: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< wifinetwork_object >

The wifinetwork_object element is used by a wifinetwork_test to define the SSID of the Wi-Fi to verify security settings. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information.

Extends: oval-def:ObjectType

Child Elements

Table 267: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
ssid	oval-def:EntityObjectStringType (1..1)	The network's SSID to check.
oval-def:filter	n/a (0..unbounded)	

< wifinetwork_state >

The wifinetwork_state element defines the Wi-Fi network settings status.

Extends: oval-def:StateType

Child Elements

Table 268: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
ssid	oval-def:EntityStateStringType (0..1)	The network's SSID.
bssid	oval-def:EntityStateStringType (0..1)	BSSID. The value is a string in the format of an Ethernet MAC address.
auth_algorithms	android-def:EntityStateWifiAuthAlgorithmType (0..unbounded)	The set of authentication protocols supported by this configuration.
group_ciphers	android-def:EntityStateWifiGroupCipherType (0..unbounded)	The set of group ciphers supported by this configuration.
key_management	android-def:EntityStateWifiKeyMgmtType (0..unbounded)	The set of key management protocols supported by this configuration.
pairwise_ciphers	android-def:EntityStateWifiPairwiseCipherType (0..unbounded)	The set of pairwise ciphers for WPA supported by this configuration.
protocols	android-def:EntityStateWifiProtocolType (0..unbounded)	The set of security protocols supported by this configuration.
hidden_ssid	oval-def:EntityStateBoolType (0..1)	This is a network that does not broadcast its SSID.
network_id	oval-def:EntityStateIntType (0..1)	The ID number that the supplicant uses to identify this network configuration entry.
priority	oval-def:EntityStateIntType (0..1)	Priority determines the preference given to a network by wpa_supplicant when choosing an access point with which to associate.
current_status	android-def:EntityStateWifiCurrentStatusType (0..1)	The current status of this network configuration entry.

< telephony_test >

The telephony_test is used to check Telephony characteristics of system.

Extends: oval-def:TestType

Child Elements

Table 269: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< telephony_object >

The telephony_object element is used by a telephony test to define those objects to evaluate based on a telephony manager state.

Extends: oval-def:ObjectType

< telephony_state >

The telephony_state element contains a single entity that is used to check the status of the telephony manager state.

Extends: oval-def:StateType

Child Elements

Table 270: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
network_type	android-def:EntityStateNetworkType (0..1)	Value indicates the radio technology(network type) currently in use, for data transmission.
sim_country_iso	oval-def:EntityStateStringType (0..1)	The ISO country code equivalent for the SIM provider's country code.
sim_operator_code	oval-def:EntityStateStringType (0..1)	The MCC+MNC(mobile country code + mobile network code) of the provider of the SIM. It contains 5 or 6 decimal digits.

== EntityStateEncryptionStatusType ==

The EntityStateEncryptionStatusType complex type restricts a string value to a specific set of values. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 271: Enumeration Values

Value	Description
ENCRYPTION_STATUS_UNSUPPORTED	Encryption is not supported
ENCRYPTION_STATUS_ACTIVE	Encryption is active.
ENCRYPTION_STATUS_INACTIVE	Encryption is supported but is not currently active.
ENCRYPTION_STATUS_ACTIVATING	Encryption is not currently active, but is currently being activated.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateKeyguardDisabledFeaturesType ==

The EntityStateKeyguardDisabledFeaturesType complex type restricts a string value to a specific set of values. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 272: Enumeration Values

Value	Description
KEYGUARD_DISABLE_FEATURES_NONE	Widgets are enabled in keyguard
KEYGUARD_DISABLE_WIDGETS_ALL	Disable all keyguard widgets
KEYGUARD_DISABLE_SECURE_CAMERA	Disable the camera on secure keyguard screens (e.g. PIN/Pattern/Password)
KEYGUARD_DISABLE_FEATURES_ALL	Disable all current and future keyguard customizations
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateNetworkType ==

The EntityStateNetworkType complex type restricts a string value to a specific set of values. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 273: Enumeration Values

Value	Description
UNKNOWN	The network type is unknown
GPRS	Current network is GPRS
EDGE	Current network is EDGE
UMTS	Current network is UMTS
CDMA	Current network is CDMA
EVDO-0	Current network is EVDO-0
EVDO-A	Current network is EVDO-A
1xRTT	Current network is 1xRTT
HSDPA	Current network is HSDPA
HSUPA	Current network is HSUPA
HSPA	Current network is HSPA
IDEN	Current network is IDEN
EVDO-B	Current network is EVDO-B
LTE	Current network is LTE
EHRPD	Current network is EHRPD
HSPAP	Current network is HSPAP

== EntityStatePasswordQualityType ==

The EntityStatePasswordQualityType complex type restricts a string value to a specific set of values. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 274: Enumeration Values

Value	Description
PASSWORD_QUALITY_ALPHABETIC	The password must contain alphabetic (or other symbol) characters
PASSWORD_QUALITY_ALPHANUMERIC	The password must contain both numeric and alphabetic (or other symbol) characters
PASSWORD_QUALITY_BIOMETRIC_WEAK	This policy allows for low-security biometric recognition technology
PASSWORD_QUALITY_COMPLEX	The password must contain at least a letter, a numerical digit, and a special symbol
PASSWORD_QUALITY_NUMERIC	The password must contain at least numeric characters
PASSWORD_QUALITY_SOMETHING	This policy requires some kind of password, but doesn't care what it is
PASSWORD_QUALITY_UNSPECIFIED	There are no password policy requirements
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateWifiAuthAlgorithmType ==

The EntityStateWifiAuthAlgorithmType complex type restricts a string value to a specific set of values that name WiFi authentication algorithms. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 275: Enumeration Values

Value	Description
LEAP	LEAP/Network EAP (only used with LEAP)
OPEN	Open System authentication (required for WPA/WPA2)
SHARED	Shared Key authentication (requires static WEP keys)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateWifiCurrentStatusType ==

The EntityStateWifiCurrentStatusType complex type restricts a string value to a specific set of values. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 276: Enumeration Values

Value	Description
CURRENT	The network we are currently connected to
ENABLED	Supplicant will not attempt to use this network
DISABLED	Supplicant will consider this network available for association
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateWifiGroupCipherType ==

The EntityStateWifiGroupCipherType complex type restricts a string value to a specific set of values that name Wi-Fi group ciphers (android.net.wifi.WifiConfiguration.GroupCipher). The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 277: Enumeration Values

Value	Description
CCMP	AES in Counter mode with CBC-MAC [RFC 3610, IEEE 802.11i/D7.0]; Constant Value: 3 (0x00000003)
TKIP	Temporal Key Integrity Protocol [IEEE 802.11i/D7.0]; Constant Value: 2 (0x00000002)
WEP104	WEP (Wired Equivalent Privacy) with 104-bit key; Constant Value: 1 (0x00000001)
WEP40	WEP (Wired Equivalent Privacy) with 40-bit key (original 802.11); Constant Value: 0 (0x00000000)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateWifiKeyMgmtType ==

The EntityStateWifiKeyMgmtType complex type restricts a string value to a specific set of values that name Wi-Fi key management schemes (from android.net.wifi.WifiConfiguration.KeyMgmt). The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 278: Enumeration Values

Value	Description
IEEE8021X	IEEE 802.1X using EAP authentication and (optionally) dynamically generated WEP keys.
NONE	WPA is not used; plaintext or static WEP could be used.
WPA_EAP	WPA using EAP authentication.
WPA_PSK	WPA pre-shared key.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateWifiPairwiseCipherType ==

The EntityStateWifiPairwiseCipherType complex type restricts a string value to a specific set of values that name Wi-Fi recognized pairwise ciphers for WPA (from android.net.wifi.WifiConfiguration.PairwiseCipher). The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 279: Enumeration Values

Value	Description
CCMP	AES in Counter mode with CBC-MAC [RFC 3610, IEEE 802.11i/D7.0]
NONE	Use only Group keys (deprecated)
TKIP	Temporal Key Integrity Protocol [IEEE802.11i/D7.0]
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateWifiProtocolType ==

The EntityStateWifiProtocolType complex type restricts a string value to a specific set of values that name Wi-Fi recognized security protocols (from android.net.wifi.WifiConfiguration.Protocol). The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 280: Enumeration Values

Value	Description
RSN	WPA2/IEEE 802.11i
WPA	WPA/IEEE 802.11i/D3.0
	The empty string value is permitted here to allow for empty elements associated with variable references.

Open Vulnerability and Assessment Language: Android System Characteristics

- Schema: Android System Characteristics
- Version: 5.11.1:1.1
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the Android specific system characteristic items found in Open Vulnerability and Assessment Language (OVAL). Each item is an extension of the standard item element defined in the Core System Characteristic Schema. Through extension, each item inherits a set of elements and attributes that are shared amongst all OVAL Items. Each item is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core System Characteristic Schema is not outlined here.

The OVAL Schema is maintained by the 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.cisecurity.org.

Item Listing

- *< appmanager_item >*
- *< bluetooth_item >*
- *< camera_item >*
- *< certificate_item >*
- *< devicesettings_item >*
- *< encryption_item >*

- *< locationservice_item >*
- *< network_item >*
- *< password_item >*
- *< systemdetails_item >*
- *< wifi_item >*
- *< wifinetwork_item >*
- *< telephony_item >*

< appmanager_item >

This item stores information about applications installed on the device.

Extends: oval-sc:ItemType

Child Elements

Table 281: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
application_name	oval-sc:EntityItemStringType (0..1)	Name of the application.
uid	oval-sc:EntityItemStringType (0..1)	Linux userid assigned to the application. (In some cases multiple applications can share a userid.)
gid	oval-sc:EntityItemStringType (0..unbounded)	One element for each group id that the application belongs to.
package_name	oval-sc:EntityItemStringType (0..1)	Name of the package.
data_directory	oval-sc:EntityItemStringType (0..1)	Data directory assigned to the application.
version	oval-sc:EntityItemStringType (0..1)	Application version.
current_status	oval-sc:EntityItemBoolType (0..1)	True if the application is enabled.
permission	oval-sc:EntityItemStringType (0..unbounded)	One element for each permission granted to the application.
native_lib_dir	oval-sc:EntityItemStringType (0..1)	Directory where the application's native libraries (if any) have been installed.
signing_certificate	oval-sc:EntityItemBinaryType (0..unbounded)	Hexadecimal string of the signing certificate corresponding with the key used to sign the application package. Only the actual signing certificate should be included, not CA certificates in the chain (if applicable).
first_install_time	oval-sc:EntityItemIntType (0..1)	Time at which the app was first installed, expressed in milliseconds since January 1, 1970 00:00:00 UTC.
last_update_time	oval-sc:EntityItemIntType (0..1)	Time at which the app was last updated, expressed in milliseconds since January 1, 1970 00:00:00 UTC.
package_file_location	oval-sc:EntityItemStringType (0..1)	From ApplicationInfo.sourceDir, the full path to the location of the publicly available parts of the application package.

< bluetooth_item >

This holds information about device Bluetooth settings.

Extends: oval-sc:ItemType

Child Elements

Table 282: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
discoverable	oval-sc:EntityItemBoolType (0..1)	True if device Bluetooth is currently in discoverable mode.
current_status	oval-sc:EntityItemBoolType (0..1)	True if device Bluetooth is currently enabled.

< camera_item >

This item is used to check camera-related information.

Extends: oval-sc:ItemType

Child Elements

Table 283: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
camera_disabled_policy	oval-sc:EntityItemBoolType (0..1)	If true, then a policy is being enforced disabling use of the camera. The policy is only available in Android 4.0 and up (and potentially on older Android devices if specifically added by the device vendor).

< certificate_item >

This item stores information about the certificates installed on the device.

Extends: oval-sc:ItemType

Child Elements

Table 284: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
trusted_certificate	oval-sc:EntityItemBinaryType (0..unbounded)	Hexadecimal string of each certificate in the OS's trusted certificate store, including both certificates installed by the system and by users. System trusted certificates that were disabled by the user are not included here.

< devicesettings_item >

This holds information about miscellaneous device settings.

Extends: oval-sc:ItemType

Child Elements

Table 285: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
adb_enabled	oval-sc:EntityItemBoolType (0..1)	True if Android Debug Bridge (USB debugging) is enabled.
allow_mock_location	oval-sc:EntityItemBoolType (0..1)	True if mock locations and location provider status can be injected into Android's Location Manager.
install_non_market_apps	oval-sc:EntityItemBoolType (0..1)	True if applications can be installed from "unknown sources".
device_admin	oval-sc:EntityItemStringType (0..unbounded)	One element per application that holds device administrator access. Contains the application's package name.
auto_time	oval-sc:EntityItemBoolType (0..1)	True if the user prefers the date and time to be automatically fetched from the network.
auto_time_zone	oval-sc:EntityItemBoolType (0..1)	True if the user prefers the time zone to be automatically fetched from the network.
usb_mass_storage_enabled	oval-sc:EntityItemBoolType (0..1)	True if USB mass storage is enabled on the device, otherwise false.

< encryption_item >

Device encryption information.

Extends: oval-sc:ItemType

Child Elements

Table 286: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
encryption_policy_enabled	oval-sc:EntityItemBoolType (0..1)	True if a policy is in place requiring the device storage to be encrypted. (android.app.admin.DevicePolicyManager.getStorageEncryption())
encryption_status	android-sc:EntityItemEncryptionStatusType (0..1)	The current status of device encryption. (android.app.admin.DevicePolicyManager.getStorageEncryptionStatus()) Either ENCRYPTION_STATUS_UNSUPPORTED, ENCRYPTION_STATUS_INACTIVE, ENCRYPTION_STATUS_ACTIVATING, or ENCRYPTION_STATUS_ACTIVE as documented in the Android SDK's DevicePolicyManager class.

< locationservice_item >

This holds information about location based service status.

Extends: oval-sc:ItemType

Child Elements

Table 287: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
gps_enabled	oval-sc:EntityItemBoolType (0..1)	A boolean value indicating whether the GPS location provider is enabled.
network_enabled	oval-sc:EntityItemBoolType (0..1)	A boolean value indicating whether the network location provider is enabled.

< network_item >

This holds information about networks configured and their preference.

Extends: oval-sc:ItemType

Child Elements

Table 288: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
airplane_mode	oval-sc:EntityItemBoolType (0..1)	True if airplane mode is enabled.
nfc_enabled	oval-sc:EntityItemBoolType (0..1)	True if NFC is enabled on the device.

< password_item >

Specific policy items associated with passwords and the device screen lock.

Extends: oval-sc:ItemType

Child Elements

Table 289: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
max_num_failed_user_auth	oval-sc:EntityItemIntType (0..1)	Maximum number of failed user authentications before device wipe. Zero means there is no policy in place.
password_hist	oval-sc:EntityItemIntType (0..1)	Specifies the length of password history maintained (passwords in the history cannot be reused). Zero means there is no policy in place.
password_quality	android-sc:EntityItemPasswordQualityType (0..1)	The current minimum required password quality required by device policy. Represented as a string corresponding with a valid Android password quality, currently one of: PASSWORD_QUALITY_ALPHABETIC PASSWORD_QUALITY_ALPHANUMERIC PASSWORD_QUALITY_BIOMETRIC_WEAK PASSWORD_QUALITY_COMPLEX PASSWORD_QUALITY_NUMERIC PASSWORD_QUALITY_SOMETHING PASSWORD_QUALITY_UNSPECIFIED
password_min_length	oval-sc:EntityItemIntType (0..1)	Minimum length of characters password must have. This constraint is only imposed if the password quality is one of PASSWORD_QUALITY_NUMERIC, PASSWORD_QUALITY_ALPHABETIC, PASSWORD_QUALITY_ALPHANUMERIC, or PASSWORD_QUALITY_COMPLEX.
password_min_letters	oval-sc:EntityItemIntType (0..1)	Minimum number of letters password must have. This constraint is only imposed if the password quality is PASSWORD_QUALITY_COMPLEX.
password_min_lowercase_letters	oval-sc:EntityItemIntType (0..1)	Minimum number of lower case letters password must have. This constraint is only imposed if the password quality is PASSWORD_QUALITY_COMPLEX.
password_min_non_letters	oval-sc:EntityItemIntType (0..1)	Minimum number of non-letter characters password must have. This constraint is only imposed if the password quality is PASSWORD_QUALITY_COMPLEX.
password_min_numeric	oval-sc:EntityItemIntType (0..1)	Minimum number of numeric characters password must have. This constraint is only imposed if the password quality is PASSWORD_QUALITY_COMPLEX.
password_min_symbols	oval-sc:EntityItemIntType (0..1)	Minimum number of symbol characters password must have. This constraint is only imposed if the password quality is PASSWORD_QUALITY_COMPLEX.
password_min_uppercase_letters	oval-sc:EntityItemIntType (0..1)	Minimum number of upper case letters password must have. This constraint is only imposed if the password quality is PASSWORD_QUALITY_COMPLEX.
password_expiration_timeout	oval-sc:EntityItemIntType (0..1)	Gets the current password expiration timeout policy, in milliseconds. Zero means there is no policy in place.
password_visible	oval-sc:EntityItemBoolType (0..1)	When true, the most recently keyed in password character is shown to the user on the screen (the previously entered characters are masked out). When false, all keyed in password characters are immediately masked out. This setting is manageable by the device user through the device settings.
active_password_sufficient	oval-sc:EntityItemBoolType (0..1)	When true, the current device password is compliant with the password policy. (If the policy was recently established, it is possible that a password compliant with the policy may not yet be in place.)
current_failed_password_attempts	oval-sc:EntityItemIntType (0..1)	The number of times the user has failed at entering a password since the last successful password entry.
screen_lock_timeout	oval-sc:EntityItemIntType (0..1)	The current policy for the highest screen lock timeout the user is allowed to specify. 0 indicates no restriction. (The user may still specify lower values in the device settings.)
keyguard_disabled_features	android-sc:EntityItemKeyguardDisabledFeaturesType (0..1)	The current policy for lockscreen widgets as retrieved by DevicePolicyManager.getKeyguardDisabledFeatures. May be

< systemdetails_item >

This item stores information about the Operating System and hardware.

Extends: oval-sc:ItemType

Child Elements

Table 290: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
hardware	oval-sc:EntityItemStringType (0..1)	The hardware model, as provided by android.os.Build.HARDWARE using the Android SDK.
manufacturer	oval-sc:EntityItemStringType (0..1)	The device manufacturer, as provided by android.os.Build.MANUFACTURER using the Android SDK.
model	oval-sc:EntityItemStringType (0..1)	The device model identifier, as provided by android.os.Build.MODEL using the Android SDK.
product	oval-sc:EntityItemStringType (0..1)	The product name, as provided by android.os.Build.PRODUCT using the Android SDK.
cpu_abi	oval-sc:EntityItemStringType (0..1)	The name of the instruction set of native code, as provided by android.os.Build.CPU_ABI using the Android SDK.
cpu_abi2	oval-sc:EntityItemStringType (0..1)	The name of the second instruction set of native code, as provided by android.os.Build.CPU_ABI2 using the Android SDK.
build_fingerprint	oval-sc:EntityItemStringType (0..1)	Build fingerprint, as provided by android.os.Build.FINGERPRINT using the Android SDK.
os_version_code_name	oval-sc:EntityItemStringType (0..1)	Operating system version code, as provided by android.os.Build.VERSION.CODENAME using the Android SDK.
os_version_build_number	oval-sc:EntityItemStringType (0..1)	Operating system build number, as provided by android.os.Build.VERSION.INCREMENTAL using the Android SDK.
os_version_release_name	oval-sc:EntityItemStringType (0..1)	Operating system release name, as provided by android.os.Build.VERSION.RELEASE using the Android SDK.
os_version_sdk_number	oval-sc:EntityItemIntType (0..1)	Operating system SDK number, as provided by android.os.Build.VERSION.SDK_INT using the Android SDK.
hardware_keystore	oval-sc:EntityItemBoolType (0..1)	True if the device provides a hardware backed cryptographic keystore (a hardware keystore prevents exporting private keys or directly exposing private keys to the OS), otherwise false.

< wifi_item >

This item holds information about general Wi-Fi settings.

Extends: oval-sc:ItemType

Child Elements

Table 291: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
wifi_status	oval-sc:EntityItemBoolType (0..1)	True if Wi-Fi is currently enabled on the device.
network_availability_notification	oval-sc:EntityItemBoolType (0..1)	True if the Wi-Fi network availability notification setting is currently enabled on the device.

< wifinetwork_item >

This item holds information about the configured Wi-Fi networks on the device.

Extends: oval-sc:ItemType

Child Elements

Table 292: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
bssid	oval-sc:EntityItemStringType (0..1)	BSSID. The value is a string in the format of an Ethernet MAC address.
ssid	oval-sc:EntityItemStringType (0..1)	The network's SSID.
auth_algorithms	android-sc:EntityItemWifiAuthAlgorithmType (0..unbounded)	The set of authentication protocols supported by this configuration.
group_ciphers	android-sc:EntityItemWifiGroupCipherType (0..unbounded)	The set of group ciphers supported by this configuration.
key_management	android-sc:EntityItemWifiKeyMgmtType (0..unbounded)	The set of key management protocols supported by this configuration.
pairwise_ciphers	android-sc:EntityItemWifiPairwiseCipherType (0..unbounded)	The set of pairwise ciphers for WPA supported by this configuration.
protocols	android-sc:EntityItemWifiProtocolType (0..unbounded)	The set of security protocols supported by this configuration.
hidden_ssid	oval-sc:EntityItemBoolType (0..1)	This is a network that does not broadcast its SSID.
network_id	oval-sc:EntityItemIntType (0..1)	The ID number that the supplicant uses to identify this network configuration entry.
priority	oval-sc:EntityItemIntType (0..1)	Priority determines the preference given to a network by wpa_supplicant when choosing an access point with which to associate.
current_status	android-sc:EntityItemWifiCurrentStatusType (0..1)	The current status of this network configuration entry, either CURRENT, DISABLED, or ENABLED per android.net.wifi.WifiConfiguration.Status.

< telephony_item >

The telephony_item element contains a single entity that is used to check the status of the telephony manager Item.

Extends: oval-sc:ItemType

Child Elements

Table 293: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
network_type	android-sc:EntityItemNetworkType (0..1)	A constant String value indicating the radio technology (network type) currently in use on the device for data transmission.
sim_country_iso	oval-sc:EntityItemStringType (0..1)	The ISO country code equivalent for the SIM provider's country code.
sim_operator_code	oval-sc:EntityItemStringType (0..1)	the MCC+MNC (mobile country code + mobile network code) of the provider of the SIM. It contains 5 or 6 decimal digits.

== EntityItemEncryptionStatusType ==

The EntityItemEncryptionStatusType complex type restricts a string value to a specific set of values. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 294: Enumeration Values

Value	Description
ENCRYPTION_STATUS_UNSUPPORTED	Encryption is not supported
ENCRYPTION_STATUS_ACTIVE	Encryption is active.
ENCRYPTION_STATUS_INACTIVE	Encryption is supported but is not currently active.
ENCRYPTION_STATUS_ACTIVATING	Encryption is not currently active, but is currently being activated.
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemKeyguardDisabledFeaturesType ==

The EntityItemKeyguardDisabledFeaturesType complex type restricts a string value to a specific set of values. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 295: Enumeration Values

Value	Description
KEYGUARD_DISABLE_FEATURES_NONE	Widgets are enabled in keyguard
KEYGUARD_DISABLE_WIDGETS_ALL	Disable all keyguard widgets
KEYGUARD_DISABLE_SECURE_CAMERA	Disable the camera on secure keyguard screens (e.g. PIN/Pattern/Password)
KEYGUARD_DISABLE_FEATURES_ALL	Disable all current and future keyguard customizations
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemNetworkType ==

The EntityItemNetworkType complex type restricts a string value to a specific set of values. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 296: Enumeration Values

Value	Description
UNKNOWN	The network type is unknown
GPRS	Current network is GPRS
EDGE	Current network is EDGE
UMTS	Current network is UMTS
CDMA	Current network is CDMA
EVDO-0	Current network is EVDO-0
EVDO-A	Current network is EVDO-A
1xRTT	Current network is 1xRTT
HSDPA	Current network is HSDPA
HSUPA	Current network is HSUPA
HSPA	Current network is HSPA
IDEN	Current network is IDEN
EVDO-B	Current network is EVDO-B
LTE	Current network is LTE
EHRPD	Current network is EHRPD
HSPAP	Current network is HSPAP

== EntityItemPasswordQualityType ==

The EntityItemPasswordQualityType complex type restricts a string value to a specific set of values. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 297: Enumeration Values

Value	Description
PASSWORD_QUALITY_ALPHABETIC	The password must contain alphabetic (or other symbol) characters
PASSWORD_QUALITY_ALPHANUMERIC	The password must contain both numeric and alphabetic (or other symbol) characters
PASSWORD_QUALITY_BIOMETRIC_WEAK	This policy allows for low-security biometric recognition technology
PASSWORD_QUALITY_COMPLEX	The password must contain at least a letter, a numerical digit, and a special symbol
PASSWORD_QUALITY_NUMERIC	The password must contain at least numeric characters
PASSWORD_QUALITY_SOMETHING	This policy requires some kind of password, but doesn’t care what it is
PASSWORD_QUALITY_UNSPECIFIED	There are no password policy requirements
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemWifiAuthAlgorithmType ==

The EntityItemWifiAuthAlgorithmType complex type restricts a string value to a specific set of values that name WiFi authentication algorithms. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 298: Enumeration Values

Value	Description
LEAP	LEAP/Network EAP (only used with LEAP)
OPEN	Open System authentication (required for WPA/WPA2)
SHARED	Shared Key authentication (requires static WEP keys)
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemWifiCurrentStatusType ==

The EntityItemWifiCurrentStatusType complex type restricts a string value to a specific set of values. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 299: Enumeration Values

Value	Description
CURRENT	The network we are currently connected to
ENABLED	Supplicant will not attempt to use this network
DISABLED	Supplicant will consider this network available for association
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemWifiGroupCipherType ==

The EntityItemWifiGroupCipherType complex type restricts a string value to a specific set of values that name Wi-Fi group ciphers. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 300: Enumeration Values

Value	Description
CCMP	AES in Counter mode with CBC-MAC [RFC 3610, IEEE 802.11i/D7.0]; Constant Value: 3 (0x00000003)
TKIP	Temporal Key Integrity Protocol [IEEE 802.11i/D7.0]; Constant Value: 2 (0x00000002)
WEP104	WEP (Wired Equivalent Privacy) with 104-bit key; Constant Value: 1 (0x00000001)
WEP40	WEP (Wired Equivalent Privacy) with 40-bit key (original 802.11); Constant Value: 0 (0x00000000)
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemWifiKeyMgmtType ==

The EntityItemWifiKeyMgmtType complex type restricts a string value to a specific set of values that name Wi-Fi key management schemes (from android.net.wifi.WifiConfiguration.KeyMgmt). The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 301: Enumeration Values

Value	Description
IEEE8021X	IEEE 802.1X using EAP authentication and (optionally) dynamically generated WEP keys.
NONE	WPA is not used; plaintext or static WEP could be used.
WPA_EAP	WPA using EAP authentication.
WPA_PSK	WPA pre-shared key.
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemWifiPairwiseCipherType ==

The EntityItemWifiPairwiseCipherType complex type restricts a string value to a specific set of values that name Wi-Fi recognized pairwise ciphers for WPA (from android.net.wifi.WifiConfiguration.PairwiseCipher). The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 302: Enumeration Values

Value	Description
CCMP	AES in Counter mode with CBC-MAC [RFC 3610, IEEE 802.11i/D7.0]
NONE	Use only Group keys (deprecated)
TKIP	Temporal Key Integrity Protocol [IEEE802.11i/D7.0]
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemWifiProtocolType ==

The EntityItemWifiProtocolType complex type restricts a string value to a specific set of values that name Wi-Fi recognized security protocols (from android.net.wifi.WifiConfiguration.Protocol). The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 303: Enumeration Values

Value	Description
RSN	WPA2/IEEE 802.11i
WPA	WPA/IEEE 802.11i/D3.0
	The empty string value is permitted here to allow for detailed error reporting.

Open Vulnerability and Assessment Language: Cisco ASA Definition

- Schema: Cisco ASA Definition
- Version: 5.11.1:1.2
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the Cisco ASA specific tests found in Open Vulnerability and Assessment Language (OVAL). Each test is an extension of the standard test element defined in the Core Definition Schema. Through extension, each test inherits a set of elements and attributes that are shared amongst all OVAL tests. Each test is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core Definition Schema is not outlined here.

The OVAL Schema is maintained by the 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.cisecurity.org.

Thanks to Omar Santos and Panos Kampanakis of Cisco for providing these tests.

Test Listing

- *< acl_test >*
- *< class_map_test >*
- *< interface_test >*
- *< line_test >*
- *< policy_map_test >*
- *< service_policy_test >*

- *< snmp_host_test >*
- *< snmp_user_test >*
- *< snmp_group_test >*
- *< tcp_map_test >*
- *< version_test >*

< acl_test >

The acl test is used to check the properties of specific output lines from an ACL configuration.

Extends: oval-def:TestType

Child Elements

Table 304: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< acl_object >

The acl_object element is used by an acl_test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An acl object consists of a an acl name and an IP version entity that is the name and the IP protocol version of the access-list to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 305: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityObjectStringType (1..1)	The name of the ACL.
ip_version	asa-def:EntityObjectAccessListIPVersionType (1..1)	The IP version of the ACL.
oval-def:filter	n/a (0..unbounded)	

< acl_state >

The acl_state element defines the different information that can be used to evaluate the result of a specific ACL configuration. This includes the name of ths ACL and the corresponding config lines. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 306: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	The name of the ACL.
ip_version	asa-def:EntityStateAccessListIPVersionType (0..1)	The IP version of the ACL (i.e. IPv4 or IPv6 or both for UACLs).
use	asa-def:EntityStateAccessListUseType (0..1)	The feature where the ACL is used.
used_in	oval-def:EntityStateStringType (0..1)	The name of where the ACL is used. For example if use is 'INTERFACE', use_in will be the name of the interface.
interface_direction	asa-def:EntityStateAccessListInterfaceDirectionType (0..1)	The direction the ACL is applied by using the access-group command. Inbound access lists apply to traffic as it enters an interface.
acl_config_lines	oval-def:EntityStateStringType (0..1)	The value returned with all config lines of the ACL.
config_line	oval-def:EntityStateStringType (0..1)	The value returned with one ACL config line at a time.

< class_map_test >

The class_map test is used to check the properties of specific output lines from an MPF class-map configuration.

Extends: oval-def:TestType

Child Elements

Table 307: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< class_map_object >

The class_map_object element is used by an class_map test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A class_map object consists of a name entity that is the name of the ASA ‘class-map’ configuration to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 308: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityObjectStringType (1..1)	The MPF class-map name.
oval-def:filter	n/a (0..unbounded)	

< class_map_state >

The class_map_state element defines the different information that can be used to evaluate the result of a specific ‘class-map’ ASA command. This includes the name, the type, the inspection type, the match type, the match commands, the policy-map or class-map it is used and the action in the policy-map. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 309: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	The name of the class-map.
type	asa-def:EntityStateClassMapType (0..1)	The type of the ‘class-map nameX type’ command.
type_inspect	asa-def:EntityStateInspectionType (0..1)	The inspection type of the class-map (‘class-map nameX type inspect’).
match_all_any	asa-def:EntityStateMatchType (0..1)	The ‘match-all’ or ‘match-any’ type of the class-map. ASA defaults to ‘match-any’.
match	oval-def:EntityStateStringType (0..1)	The ‘match’ commands in the class-map.
used_in_class_map	oval-def:EntityStateStringType (0..1)	The name of the class-map (for nested class-maps) that this class-map is used in.
used_in_policy_map	oval-def:EntityStateStringType (0..1)	The name of the policy-map that this class-map is used in.
policy_map_action	oval-def:EntityStateStringType (0..1)	The command that identifies the action for the class. For example that could be ‘inspect protocolX’, ‘drop’ or ‘police 1000’ or ‘set connection advanced-options tcpmapX’.

< interface_test >

The interface test is used to check for the existence of a particular interface on the Cisco ASA device. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a interface_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 310: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< interface_object >

The interface_object element is used by an interface_test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An interface_object consists of a name entity that is the name of the ASA interface to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 311: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityObjectStringType (1..1)	The interface name.
oval-def:filter	n/a (0..unbounded)	

< interface_state >

The interface_state element defines the different information that can be used to evaluate the result of a specific ASA interface. This includes the name, status, and address information about the interface. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 312: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	The interface name.
proxy_arp	oval-def:EntityStateBoolType (0..1)	Proxy arp enabled on the interface. The default is true.
shutdown	oval-def:EntityStateBoolType (0..1)	Interface is shut down.
hardware_addr	oval-def:EntityStateStringType (0..1)	The interface hardware (MAC) address.
ipv4_address	oval-def:EntityStateIPAddressStringType (0..1)	The interface IPv4 address and mask. This element should only allow 'ipv4_address' of the oval:SimpleDatatypeEnumeration.
ipv6_address	oval-def:EntityStateIPAddressStringType (0..1)	The interface IPv6 address and mask. This element should only allow 'ipv6_address' of the oval:SimpleDatatypeEnumeration.
ipv4_access_list	oval-def:EntityStateStringType (0..1)	The ingress or egress IPv4 ACL name applied on the interface.
ipv6_access_list	oval-def:EntityStateStringType (0..1)	The ingress or egress IPv6 ACL name applied on the interface.
ipv4_v6_access_list	oval-def:EntityStateStringType (0..1)	The ingress or egress UACL name applied on the interface.
crypto_map	oval-def:EntityStateStringType (0..1)	The crypto map name applied to the interface.
ipv4_urpf_command	oval-def:EntityStateStringType (0..1)	The IPv4 uRPF command under the interface.
ipv6_urpf_command	oval-def:EntityStateStringType (0..1)	The IPv6 uRPF command under the interface.
urpf_command (Deprecated)	oval-def:EntityStateStringType (0..1)	The uRPF command under the interface.

< line_test >

The line_test is used to check the properties of specific output lines from a SHOW command, such as SHOW RUNNING-CONFIG. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a line_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 313: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< line_object >

The line_object element is used by a line_test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A line object consists of a show_subcommand entity that is the name of a SHOW sub-command to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 314: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
show_subcommand	oval-def:EntityObjectStringType (1..1)	The name of a SHOW sub-command.
oval-def:filter	n/a (0..unbounded)	

< line_state >

The line_state element defines the different information that can be used to evaluate the result of a specific SHOW sub-command. This includes the name of ths sub-command and the corresponding config line. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 315: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
show_subcommand	oval-def:EntityStateStringType (0..1)	The name of the SHOW sub-command.
config_line	oval-def:EntityStateStringType (0..1)	The value returned from by the specified SHOW sub-command.

< policy_map_test >

The policy_map test is used to check the properties of specific output lines from an policy-map ASA configuration.

Extends: oval-def:TestType

Child Elements

Table 316: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< policy_map_object >

The policy_map_object element is used by an policy_map test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A policy_map object consists of a name entity that is the name of the ASA 'policy-map' configuration to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 317: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityObjectStringType (1..1)	The MPF policy-map name.
oval-def:filter	n/a (0..unbounded)	

< policy_map_state >

The policy_map_state element defines the different information that can be used to evaluate the result of a 'policy-map' ASA configuration. This includes the policy-map name, the inspection type, the paremeters, the match and action commands, the policy-map it is used in and the service-policy that applies it. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 318: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	The policy-map name.
type_inspect	oval-def:EntityStateInspectionType (0..1)	The inspection type of the class-map.
parameters	oval-def:EntityStateStringType (0..1)	The parameter commands of the policy-map.
match_action	oval-def:EntityStateStringType (0..1)	The in-line match command and the action in the policy-map seperated by delimeter '_-_'. For example, a type inspect policy-map could have 'match body regex regexnameX' and the action be 'drop'. Then this element would be 'body regex **regexnameX_**-_drop'.
used_in	oval-def:EntityStateStringType (0..1)	The name of policy-map that includes the policy-map('policy-map type inspect' in this case) or the service-policy that applies the policy-map (non 'type inspect' in this case). For example, the former could be when a http inspection policy-map policymapnameX is used in a policy-map policymapnameY as its 'inspect http policymapnameX' command. The latter could be when policymapnameY is applied globally with 'service-policy policymapnameY global'. There is no chance where a policy-map can be used in both a policy-map and a service policy at the same time.

< service_policy_test >

The service_policy test is used to check the properties of specific output lines from an MPF service-policy configuration.

Extends: oval-def:TestType

Child Elements

Table 319: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< service_policy_object >

The service_policy_object element is used by an service_policy test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A service_policy object consists of a name entity that is the name of the ASA 'service-policy' configurate to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 320: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityObjectStringType (1..1)	The MPF service-policy name.
oval-def:filter	n/a (0..unbounded)	

< service_policy_state >

The service_policy_state element defines the different information that can be used to evaluate service-policy ASA configuration. This includes the service-policy name, where it is applied and the interface it is applied (if applicable). Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 321: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	The service-policy name.
applied	asa-def:EntityStateApplyServicePolicyType (0..1)	Where he service-policy is applied.
interface	oval-def:EntityStateStringType (0..1)	The interface the service-policy is applied (of the 'applied' element has value "INTERFACE').

< snmp_host_test >

The snmp_host test is used to check the properties of specific output lines from an SNMP configuration.

Extends: oval-def:TestType

Child Elements

Table 322: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< snmp_host_object >

The snmp_host_object element is used by an snmp_host test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A snmp_host object consists of a host entity that is the host of the 'snmp host' ASA command to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 323: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
host	oval-def:EntityObjectStringType (1..1)	The SNMP host address or hostname.
oval-def:filter	n/a (0..unbounded)	

< snmp_host_state >

The snmp_host_state element defines the different information that can be used to evaluate the result of a specific 'snmp host' ASA command. This includes the host and the corresponding options. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 324: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
interface	oval-def:EntityStateStringType (0..1)	The interface configured for the host.
host	oval-def:EntityStateStringType (0..1)	The SNMP host address or hostname.
snmpv3_user	oval-def:EntityStateStringType (0..1)	The community SNMPv3 user configured for the host.
version	asa-def:EntityStateSNMPVersionStringType (0..1)	The SNMP version.
poll	oval-def:EntityStateBoolType (0..1)	SNMP polls enabled for the host.
traps	oval-def:EntityStateBoolType (0..1)	SNMP traps enabled for the host.
udp_port	oval-def:EntityStateIntType (0..1)	SNMP port configured for the host.

< snmp_user_test >

The snmp_user test is used to check the properties of specific output lines from an SNMP user configuration.

Extends: oval-def:TestType

Child Elements

Table 325: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< snmp_user_object >

The snmp_user_object element is used by an snmp_user test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A snmp_user object consists of a name entity that is the name of the SNMP user to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 326: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityObjectStringType (1..1)	The SNMP user name.
oval-def:filter	n/a (0..unbounded)	

< snmp_user_state >

The snmp_user_state element defines the different information that can be used to evaluate the result of a specific 'show snmp-serveruser' ASA command. This includes the user name and the corresponding options. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 327: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	The SNMP user name.
group	oval-def:EntityStateStringType (0..1)	The SNMP group the user belongs to.
priv	asa-def:EntityStateSNMPPrivStringType (0..1)	The SNMP encryption type for the user (for SNMPv3).
auth	asa-def:EntityStateSNMPAuthStringType (0..1)	The SNMP authentication type for the user (for SNMPv3).

< snmp_group_test >

The snmp_group test is used to check the properties of specific output lines from an SNMP group configuration.

Extends: oval-def:TestType

Child Elements

Table 328: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< snmp_group_object >

The snmp_group_object element is used by an snmp_group test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A snmp_group object consists of a name entity that is the name of the SNMP group to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 329: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityObjectStringType (1..1)	The SNMP group name.
oval-def:filter	n/a (0..unbounded)	

< snmp_group_state >

The snmp_group_state element defines the different information that can be used to evaluate the result of a specific 'snmp-server group' ASA command. This includes the user name and the corresponding options. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 330: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	The SNMP group name.
snmpv3_sec_level	asa-def:EntityStateSNMPSecLevelStringType (0..1)	The SNMPv3 security configured for the group.

< tcp_map_test >

The tcp_map test is used to check the properties of specific output lines from a tcp-map ASA configuration.

Extends: oval-def:TestType

Child Elements

Table 331: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< tcp_map_object >

The tcp-map_object element is used by an tcp_map test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A service_policy object consists of a name entity that is the name of the ASA 'tcp-map' configuration to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 332: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityObjectStringType (1..1)	The MPF tcp-map name.
oval-def:filter	n/a (0..unbounded)	

< tcp_map_state >

The tcp_map_state element defines the different information that can be used to evaluate the result of a specific 'tcp-map' ASA configuration. This includes the tcp-map name and its configured options. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 333: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	The tcp-map name.
options	oval-def:EntityStateStringType (0..1)	The configured commends in the tcp-map. These could include TCP options, flags and other options of the tcp-map.

< version_test >

The version test is used to check the version of the ASA operating system. It is based off of the SHOW VERSION command. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a version_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 334: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< version_object >

The version_object element is used by a version test to define the different version information associated with a ASA system. There is actually only one object relating to version and this is the system as a whole. Therefore, there are no child entities defined. Any OVAL Test written to check version will reference the same version_object which is basically an empty object element.

Extends: oval-def:ObjectType

< version_state >

The version_state element defines the version information held within a Cisco ASA software release. The asa_release element specifies the whole ASA version information. The asa_major_release, asa_minor_release and asa_build elements specify seperated parts of ASA software version information. For instance, if the ASA version is 8.4(2.3)49, then asa_release is 8.4(2.3)49, asa_major_release is 8.4, asa_minor_release is 2.3 and asa_build is 49. See the SHOW VERSION command within ASA for more information.

Extends: oval-def:StateType

Child Elements

Table 335: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
asa_release	oval-def:EntityStateStringType (0..1)	The asa_release element specifies the whole ASA version information.
asa_major_release	oval-def:EntityStateVersionType (0..1)	The asa_major_release is the dotted version that starts a version string. For example the asa_release 8.4(2.3)49 has a asa_major_release of 8.4.
asa_minor_release	oval-def:EntityStateVersionType (0..1)	The asa_minor_release is the dotted version that starts a version string. For example the asa_release 8.4(2.3)49 has a asa_minor_release of 2.3.
asa_build	oval-def:EntityStateIntType (0..1)	The asa_build is an integer. For example the asa_release 8.4(2.3)49 has a asa_build of 49.

== EntityObjectAccessListIPVersionType ==

The EntityObjectAccessListIPVersionType complex type restricts a string value to a specific set of values: IPV4, IPV6 or IPV4_V6 (both). These values describe if an ACL is for IPv4 or IPv6 or both for UACLs in a Cisco ASA configuration. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityObjectStringType

Table 336: Enumeration Values

Value	Description
IPV4	(No Description)
IPV6	(No Description)
IPV4_V6	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateAccessListIPVersionType ==

The EntityStateAccessListIPVersionType complex type restricts a string value to a specific set of values: IPV4, IPV6 or IPV4_V6 (both). These values describe if an ACL is for IPv4 or IPv6 or both for UACLs in a Cisco ASA configuration. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 337: Enumeration Values

Value	Description
IPV4	(No Description)
IPV6	(No Description)
IPV4_V6	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateAccessListUseType ==

The EntityStateAccessListUseType complex type restricts a string value to a specific set of values: INTERFACE, INTERFACE_CP (control plane interface ACL), CRYPTO_MAP_MATCH, CLASS_MAP_MATCH, ROUTE_MAP_MATCH, IGMP_FILTER, NONE. These values describe the ACL use in a Cisco ASA configuration. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 338: Enumeration Values

Value	Description
INTERFACE	(No Description)
INTERFACE_CP	(No Description)
CRYPTO_MAP_MATCH	(No Description)
CLASS_MAP_MATCH	(No Description)
ROUTE_MAP_MATCH	(No Description)
IGMP_FILTER	(No Description)
NONE	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateAccessListInterfaceDirectionType ==

The EntityStateAccessListInterfaceDirectionType complex type restricts a string value to a specific set of values: IN, OUT. These values describe the inbound or outbound ACL direction on an interface in a Cisco ASA configuration. These values are defined with the access-group command. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 339: Enumeration Values

Value	Description
IN	(No Description)
OUT	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateClassMapType ==

The EntityStateClassMapType complex type restricts a string value to a specific set of values: INSPECT, REGEX, MANAGEMENT. These values describe the MPF class-map types in Cisco ASA MPF configurations. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 340: Enumeration Values

Value	Description
INSPECT	(No Description)
REGEX	(No Description)
MANAGEMENT	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateInspectionType ==

The EntityStateInspectionType complex type restricts a string value to a specific set of values. These values describe the MPF inspection types of class-map and policy-map configurations in Cisco ASA MPF configurations. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 341: Enumeration Values

Value	Description
DCERPC	(No Description)
DNS	(No Description)
ESMTP	(No Description)
FTP	(No Description)
GTP	(No Description)
H323	(No Description)
HTTP	(No Description)
IM	(No Description)
IPV6	(No Description)
MGCP	(No Description)
NETBIOS	(No Description)
RADIUS-ACCOUNTING	(No Description)
RTSP	(No Description)
SCANSAFE	(No Description)
SIP	(No Description)
SKINNY	(No Description)
SNMP	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateApplyServicePolicyType ==

The EntityStateApplyServicePolicyType complex type restricts a string value to a specific set of values: GLOBAL, INTERFACE. These values describe where a service-policy is applied in a Cisco ASA MPF configuration. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 342: Enumeration Values

Value	Description
GLOBAL	(No Description)
INTERFACE	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateMatchType ==

The EntityStateMatchType complex type restricts a string value to a specific set of values: ANY, ALL. These values describe the match type of a class-map in a Cisco ASA MPF configuration. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 343: Enumeration Values

Value	Description
ANY	(No Description)
ALL	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateSNMPVersionStringType ==

The EntityStateSNMPVersionStringType complex type restricts a string value to a specific set of values: 1, 2c, 3. These values describe the SNMP version in a Cisco ASA configuration. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 344: Enumeration Values

Value	Description
1	(No Description)
2C	(No Description)
3	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateSNMPSecLevelStringType ==

The EntityStateSNMPSecLevelStringType complex type restricts a string value to a specific set of values: PRIV, AUTH, NO_AUTH. These values describe the SNMP security level (encryption, Authentication, None) in a Cisco ASA SNMPv3 related configurations. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 345: Enumeration Values

Value	Description
PRIV	(No Description)
AUTH	(No Description)
NO_AUTH	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateSNMPAuthStringType ==

The EntityStateSNMPAuthStringType complex type restricts a string value to a specific set of values: MD5, SHA. These values describe the authentication algorithm in a Cisco ASA SNMPv3 related configurations. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 346: Enumeration Values

Value	Description
MD5	(No Description)
SHA	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateSNMPPrivStringType ==

The EntityStateSNMPPrivStringType complex type restricts a string value to a specific set of values: DES, 3DES, AES128, AES192, and AES256. These values describe the encryption algorithm in a Cisco ASA SNMPv3 related configurations. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 347: Enumeration Values

Value	Description
DES	(No Description)
3DES	(No Description)
AES128	(No Description)
AES192	(No Description)
AES256	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

Open Vulnerability and Assessment Language: Cisco ASA System Characteristics

- Schema: Cisco ASA System Characteristics
- Version: 5.11.1:1.2
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the Cisco ASA specific system characteristic items found in Open Vulnerability and Assessment Language (OVAL). Each item is an extension of the standard item element defined in the Core System Characteristic Schema. Through extension, each item inherits a set of elements and attributes that are shared amongst all OVAL Items. Each item is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core System Characteristic Schema is not outlined here.

The OVAL Schema is maintained by the 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.cisecurity.org.

Thanks to Omar Santos and Panos Kampanakis of Cisco for providing these tests.

Item Listing

- *< acl_item >*
- *< class_map_item >*
- *< interface_item >*
- *< line_item >*
- *< policy_map_item >*
- *< service_policy_item >*
- *< snmp_host_item >*
- *< snmp_user_item >*
- *< snmp_group_item >*
- *< tcp_map_item >*
- *< version_item >*

< acl_item >

Stores command that are part of a asa configuration section. For example all configuration lines under an interface. It should not store configurations for configs that already have a separate item. For example OSPF has a router item and should not also be stored in a acl_item.

Extends: oval-sc:ItemType

Child Elements

Table 348: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	Element with the name of the ACL.
ip_version	asa-sc:EntityItemAccessListIPVersionType (0..1)	Element with the IP version of the ACL.
use	asa-sc:EntityItemAccessListUseType (0..1)	Element with the feature where the ACL is used. If the same ACL is applied in more than one feature (i.e interface and crypto map), multiple items needs to be created.
used_in	oval-sc:EntityItemStringType (0..1)	Element with the name of where the ACL is used. For example if use is 'INTERFACE', use_in will be the name of the interface. If the same ACL is applied in more than one feature (i.e interface and crypto map), multiple items needs to be created.
interface_direction	asa-sc:EntityItemAccessListInterfaceDirectionType (0..1)	Element with the direction the ACL is applied to an interface using the access-group command.
acl_config_lines	oval-sc:EntityItemStringType (0..1)	Element with the value returned with all config lines of the ACL.
config_line	oval-sc:EntityItemStringType (0..unbounded)	Element with the value returned with one ACL config line at a time.

< class_map_item >

Stores information about the MPF class-map configuration in ASA. That information includes the name, the type, the inspection type, the match type, the match commands, the policy-map or class-map it is used and the action in the policy-map.

Extends: oval-sc:ItemType

Child Elements

Table 349: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	element with the name of the class-map.
type	asa-sc:EntityItemClassMapType (0..1)	Element with the type of the 'class-map nameX type' command.
type_inspect	asa-sc:EntityItemInspectionType (0..1)	Element with the inspection type of the class-map ('class-map type inspect' command).
match_all_any	asa-sc:EntityItemMatchType (0..1)	Element with the 'match-all' or 'match-any' type of the class-map. ASA's defaults to 'match-any'.
match	oval-sc:EntityItemStringType (0..unbounded)	Element with the match command in the class-map.
used_in_class_map	oval-sc:EntityItemStringType (0..unbounded)	Element with the name of the class-map (for nested class-maps) that this class-map is used in.
used_in_policy_map	oval-sc:EntityItemStringType (0..1)	Element with the name of the policy-map that this class-map is used in.
policy_map_action	oval-sc:EntityItemStringType (0..unbounded)	Element with the command that identifies the action for the class. For example that could be 'inspect protocolX', 'drop' or 'police 1000' or 'set connection advanced-options tcpmapX'.

< interface_item >

Stores information about interfaces on an Cisco ASA device.

Extends: oval-sc:ItemType

Child Elements

Table 350: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	Element with the interface name.
proxy_arp	oval-sc:EntityItemBoolType (0..1)	Element that is true if the proxy_arp command is enabled on the interface. The default is true.
shutdown	oval-sc:EntityItemBoolType (0..1)	Element that is true if the interface is shut down. The default is false.
hardware_addr	oval-sc:EntityItemStringType (0..1)	Element with the interface hardware (MAC) address.
ipv4_address	oval-sc:EntityItemIPAddressStringType (0..1)	Element with the interface IPv4 address and mask. This element should only allow 'ipv4_address' of the oval:SimpleDatatypeEnumeration.
ipv6_address	oval-sc:EntityItemIPAddressStringType (0..unbounded)	Element with the interface IPv6 address and mask. This element should only allow 'ipv6_address' of the oval:SimpleDatatypeEnumeration.
ipv4_access_list	oval-sc:EntityItemStringType (0..2)	Element with the ingress or egress IPv4 ACL name applied on the interface.
ipv6_access_list	oval-sc:EntityItemStringType (0..2)	Element with the ingress or egress IPv6 ACL name applied on the interface.
ipv4_v6_access_list	oval-sc:EntityItemStringType (0..2)	Element with the ingress or egress UACL name applied on the interface.
crypto_map	oval-sc:EntityItemStringType (0..1)	Element with the crypto map name applied to the interface.
ipv4_urpf_command	oval-sc:EntityItemStringType (0..1)	Element with the uRPF command for IPv4 under the interface.
ipv6_urpf_command	oval-sc:EntityItemStringType (0..1)	Element with the uRPF command for IPv6 under the interface.
urpf_command (Deprecated)	oval-sc:EntityItemStringType (0..1)	Element with the uRPF command under the interface.

< line_item >

Stores the configuration information associated with the evaluation of a SHOW sub-command on Cisco ASA. This includes the name of ths sub-command and the corresponding config line.

Extends: oval-sc:ItemType

Child Elements

Table 351: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
show_subcommand	oval-sc:EntityItemStringType (0..1)	The name of the SHOW sub-command.
config_line	oval-sc:EntityItemStringType (0..1)	The value returned from by the specified SHOW sub-command.

< policy_map_item >

Stores information about a policy-map configuration in ASA. That information includes the policy-map name, the inspection type, the paremeters, the match and action commands, the policy-map it is used in and the service-policy that applies it.

Extends: oval-sc:ItemType

Child Elements

Table 352: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	Element with the policy-map name.
type_inspect	oval-sc:EntityItemInspectionType (0..1)	Element with the inspection type of the class-map.
parameters	oval-sc:EntityItemStringType (0..unbounded)	Element with the parameter commands of the policy-map.
match_action	oval-sc:EntityItemStringType (0..unbounded)	Element with the in-line match command and the action in the policy-map seperated by delimeter ‘_-_’. For example an http inspect policy-map could have ‘match body regex regexnameX’ and the action be ‘drop’. Then this element would be ‘body regex regexnameX_-_drop’.
used_in	oval-sc:EntityItemStringType (0..1)	Element with the name of policy-map that includes the policy-map(‘policy-map type inspect’ in this case) or the serice-policy that applies the policy-map (non ‘type inspect’ in this case). For example, the former could be when a http inspection policy-map policymapnameX is used in a policy-map policymapnameY as its ‘inspect http policymapnameX’ command. The latter could be when policymapnameY is applied globally with ‘service-policy policymapnameY global’. There is no chance where a policy-map can be used in both a policy-map and a service policy at the same time.

< service_policy_item >

Stores information about an MPF service-policy configuration in ASA. That information includes the service-policy name, where it is applied and the interface it is applied (if applicable).

Extends: oval-sc:ItemType

Child Elements

Table 353: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	Element with the service-policy name.
applied	asa-sc:EntityItemApplyServicePolicyType (0..1)	Element with where the service-policy is applied.
interface	oval-sc:EntityItemStringType (0..1)	Element with the interface the service-policy is applied (of the 'applied' element has value "INTERFACE').

< snmp_host_item >

Stores information about the SNMP host configuration in ASA. That information includes the host, the community or user strings, the SNMP version, the snmp security (if the SNMP version is SNMPv3) and the SNMP traps.

Extends: oval-sc:ItemType

Child Elements

Table 354: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
interface	oval-sc:EntityItemStringType (0..1)	Element with the interface configured for the host.
host	oval-sc:EntityItemStringType (0..1)	Element with the SNMP host address or hostname.
snmpv3_user	oval-sc:EntityItemStringType (0..1)	Element with the community sting or SNMPv3 user configured for the host.
version	asa-sc:EntityItemSNMPVersionStringType (0..1)	Element with the SNMP version.
poll	oval-sc:EntityItemBoolType (0..1)	Element used for when the SNMP polls are enabled for the host.
traps	oval-sc:EntityItemBoolType (0..1)	Element used for when the SNMP polls are enabled for the host.
udp_port	oval-sc:EntityItemIntType (0..1)	Element used for the SNMP port configured for the host.

< snmp_user_item >

Stores information about an SNMP user configuration in ASA. That information includes the user name, the SNMP group he belongs to, the SNMP version, the IPv4 or IPv6 ACL it is applied to, the Security Level and the Authentication type that apply to the user (for SNMPv3).

Extends: oval-sc:ItemType

Child Elements

Table 355: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	Element with the SNMP user name.
group	oval-sc:EntityItemStringType (0..1)	Element with the SNMP group the user belongs to.
priv	asa-sc:EntityItemSNMPPrivStringType (0..1)	Element with the SNMP encryption type for the user (for SNMPv3).
auth	asa-sc:EntityItemSNMPAuthStringType (0..1)	Element with the SNMP authentication type for the user (for SNMPv3).

< snmp_group_item >

Stores information about an SNMP group configuration in ASA. That information includes the group name, the SNMP version, the IPv4 or IPv6 ACL it is applied to and the read, write and/or notify views applied to the group.

Extends: oval-sc:ItemType

Child Elements

Table 356: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	Element with the SNMP group name.
snmpv3_sec_level	asa-sc:EntityItemSNMPSecLevelStringType (0..1)	Element with the SNMPv3 security configure for the group.

< tcp_map_item >

Stores information about MPF tcp-map configuration in ASA. That information includes the tcp-map name and its configured options.

Extends: oval-sc:ItemType

Child Elements

Table 357: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	Element with the tcp-map name.
options	oval-sc:EntityItemStringType (0..unbounded)	Element with the configured commends in the tcp-map. These could include TCP options, flags and other options of the tcp-map.

< version_item >

Stores the version information held within a Cisco ASA software release. The asa_release element specifies the whole ASA version information. The asa_major_release, asa_minor_release and asa_build elements specify seperated parts of ASA software version information. For instance, if the ASA version is 8.4(2.3)49, then asa_release is 8.4(2.3)49, asa_major_release is 8.4, asa_minor_release is 2.3 and asa_build is 49. See the SHOW VERSION command within ASA for more information.

Extends: oval-sc:ItemType

Child Elements

Table 358: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
asa_release	oval-sc:EntityItemStringType (0..1)	The asa_release element specifies the whole ASA version information.
asa_major_release	oval-sc:EntityItemVersionType (0..1)	The asa_major_release is the dotted version that starts a version string. For example the asa_release 8.4(2.3)49 has a asa_major_release of 8.4.
asa_minor_release	oval-sc:EntityItemVersionType (0..1)	The asa_minor_release is the dotted version that starts a version string. For example the asa_release 8.4(2.3)49 has a asa_minor_release of 2.3.
asa_build	oval-sc:EntityItemIntType (0..1)	The asa_build is an integer. For example the asa_release 8.4(2.3)49 has a asa_build of 49.

== EntityItemAccessListIPVersionType ==

The EntityItemAccessListIPVersionType complex type restricts a string value to a specific set of values: IPV4, IPV6 or IPV4_V6 (both). These values describe if an ACL is for IPv4 or both for UACLs or IPv6 in a Cisco asa configuration. The empty string is also allowed to support empty element associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 359: Enumeration Values

Value	Description
IPV4	(No Description)
IPV6	(No Description)
IPV4_V6	(No Description)
	The empty string value is permitted here to allow for empty elements associated with error conditions.

== EntityItemAccessListUseType ==

The EntityItemAccessListUseType complex type restricts a string value to a specific set of values: INTERFACE, INTERFACE_CP (control plane interface ACL), CRYPTO_MAP_MATCH, CLASS_MAP_MATCH, ROUTE_MAP_MATCH, IGMP_FILTER, NONE. These values describe the ACL use in a Cisco asa configuration. The empty string is also allowed to support empty element associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 360: Enumeration Values

Value	Description
INTERFACE	(No Description)
INTERFACE_CP	(No Description)
CRYPTO_MAP_MATCH	(No Description)
CLASS_MAP_MATCH	(No Description)
ROUTE_MAP_MATCH	(No Description)
IGMP_FILTER	(No Description)
NONE	(No Description)
	The empty string value is permitted here to allow for empty elements associated with error conditions.

== EntityItemAccessListInterfaceDirectionType ==

The EntityItemAccessListInterfaceDirectionType complex type restricts a string value to a specific set of values: IN, OUT. These values describe the inbound or outbound ACL direction on an interface in a Cisco ASA configuration. The empty string is also allowed to support empty element associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 361: Enumeration Values

Value	Description
IN	(No Description)
OUT	(No Description)
	The empty string value is permitted here to allow for empty elements associated with error conditions.

== EntityItemClassMapType ==

The EntityItemClassMapType complex type restricts a string value to a specific set of values: INSPECT, REGEX, MANAGEMENT. These values describe the MPF class-map types in Cisco ASA MPF configurations. The empty string is also allowed to support empty element associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 362: Enumeration Values

Value	Description
INSPECT	(No Description)
REGEX	(No Description)
MANAGEMENT	(No Description)
	The empty string value is permitted here to allow for empty elements associated with error conditions.

== EntityItemInspectionType ==

The EntityItemInspectionType complex type restricts a string value to a specific set of values. These values describe the MPF inspection types of class-map and policy-map configurations in Cisco ASA MPF configurations. The empty string is also allowed to support empty element associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 363: Enumeration Values

Value	Description
DCERPC	(No Description)
DNS	(No Description)
ESMTP	(No Description)
FTP	(No Description)
GTP	(No Description)
H323	(No Description)
HTTP	(No Description)
IM	(No Description)
IPV6	(No Description)
MGCP	(No Description)
NETBIOS	(No Description)
RADIUS-ACCOUNTING	(No Description)
RTSP	(No Description)
SCANSAFE	(No Description)
SIP	(No Description)
SKINNY	(No Description)
SNMP	(No Description)
	The empty string value is permitted here to allow for empty elements associated with error conditions.

== EntityItemApplyServicePolicyType ==

The EntityItemApplyServicePolicyType complex type restricts a string value to a specific set of values: GLOBAL, INTERFACE. These values describe where a service-policy is applied in a Cisco ASA MPF configuration. The empty string is also allowed to support empty element associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 364: Enumeration Values

Value	Description
GLOBAL	(No Description)
INTERFACE	(No Description)
	The empty string value is permitted here to allow for empty elements associated with error conditions.

== EntityItemMatchType ==

The EntityItemMatchType complex type restricts a string value to a specific set of values: ANY, ALL. These values describe the match type of a class-map in a Cisco ASA MPF configuration. The empty string is also allowed to support empty element associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 365: Enumeration Values

Value	Description
ANY	(No Description)
ALL	(No Description)
	The empty string value is permitted here to allow for empty elements associated with error conditions.

== EntityItemSNMPVersionStringType ==

The EntityItemSNMPVersionStringType complex type restricts a string value to a specific set of values: 1, 2c, 3. These values describe the SNMP version in a Cisco ASA configuration. The empty string is also allowed to support empty element associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 366: Enumeration Values

Value	Description
1	(No Description)
2C	(No Description)
3	(No Description)
	The empty string value is permitted here to allow for empty elements associated with error conditions.

== EntityItemSNMPSecLevelStringType ==

The EntityItemSNMPSecLevelStringType complex type restricts a string value to a specific set of values: PRIV, AUTH, NO_AUTH. These values describe the SNMP security level (encryption, Authentication, None) in a Cisco ASA SNMPv3 related configurations. The empty string is also allowed to support empty element associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 367: Enumeration Values

Value	Description
PRIV	(No Description)
AUTH	(No Description)
NO_AUTH	(No Description)
	The empty string value is permitted here to allow for empty elements associated with error conditions.

== EntityItemSNMPAuthStringType ==

The EntityItemSNMPAuthStringType complex type restricts a string value to a specific set of values: MD5, SHA. These values describe the authentication algorithm in a Cisco ASA SNMPv3 related configurations. The empty string is also allowed to support empty element associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 368: Enumeration Values

Value	Description
MD5	(No Description)
SHA	(No Description)
	The empty string value is permitted here to allow for empty elements associated with error conditions.

== EntityItemSNMPPrivStringType ==

The EntityItemSNMPPrivStringType complex type restricts a string value to a specific set of values: DES, 3DES, AES128, AES192, and AES256. These values describe the encryption algorithm in a Cisco ASA SNMPv3 related configurations. The empty string is also allowed to support empty element associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 369: Enumeration Values

Value	Description
DES	(No Description)
3DES	(No Description)
AES128	(No Description)
AES192	(No Description)
AES256	(No Description)
	The empty string value is permitted here to allow for empty elements associated with error conditions.

Open Vulnerability and Assessment Language: CatOS Definition

- Schema: CatOS Definition
- Version: 5.11.1:1.1
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the Cisco CatOS specific tests found in Open Vulnerability and Assessment Language (OVAL). Each test is an extension of the standard test element defined in the Core Definition Schema. Through extension, each test inherits a set of elements and attributes that are shared amongst all OVAL tests. Each test is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core Definition Schema is not outlined here

This schema was originally developed by Yuzheng Zhou and Eric Grey at Hewlett-Packard. The OVAL Schema is maintained by the 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.cisecurity.org.

Test Listing

- *< line_test >*
- *< module_test >*
- *< version55_test >*
- *< version_test >* (Deprecated)

< line_test >

The line_test is used to check the properties of specific output lines from a SHOW command, such as show running-config. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a line_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 370: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< line_object >

The line_object element is used by a line_test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A line_object consists of a show_subcommand entity that is the name of a SHOW sub-command to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 371: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
show_subcommand	oval-def:EntityObjectStringType (1..1)	The name of a SHOW sub-command.
oval-def:filter	n/a (0..unbounded)	

< line_state >

The line_state element defines the different information that can be used to evaluate the result of a specific SHOW sub-command. This includes the name of ths sub-command and the corresponding config line. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 372: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
show_subcommand	oval-def:EntityStateStringType (0..1)	The name of the SHOW sub-command.
config_line	oval-def:EntityStateStringType (0..1)	The value returned from by the specified SHOW sub-command.

< module_test >

The module test reveals module information in Cisco Catalyst switches. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a module_object and the optional state element specifies the metadata to check.

The module_test is based off the SHOW MODULE command. Having a separate module_test, as opposed to a general command_test, enables running an evaluation based on OVAL without having interactive command access to the device.

Extends: oval-def:TestType

Child Elements

Table 373: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< module_object >

The module_object element is used by a module test to specify the module to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions schema.

A module object consists of a single module_number entity that identifies the module to be used.

Extends: oval-def:ObjectType

Child Elements

Table 374: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
module_number	oval-def:EntityObjectIntType (1..1)	A number that identifies the a specific module.
oval-def:filter	n/a (0..unbounded)	

< module_state >

The module_state element defines the module information held within a Cisco Catalyst switch. The module_number, type, and model element specifies the number, type and model of the module respectively. The software_major_release, software_individual_release and software_version_id elements specify the software version information of the module. For instance, if the software version is 8.5(4c)GLX, then software_major_release is 8.5GLX, software_individual_release is 4 and software_version_id is c. Similarly, the hardware_major_release, hardware_individual_release, firmware_major_release and firmware_individual_release elements reveal the hardware and firmware version information of the module.

Extends: oval-def:StateType

Child Elements

Table 375: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
module_number	oval-def:EntityStateIntType (0..1)	A number that identifies the a specific module.
type	oval-def:EntityStateStringType (0..1)	The type of module.
model	oval-def:EntityStateStringType (0..1)	The model of a module.
software_major_release	oval-def:EntityStateVersionType (0..1)	The major relase of the software of a module to check for.
software_individual_release	oval-def:EntityStateIntType (0..1)	The individual release of the software of the module to check for.
software_version_id	oval-def:EntityStateStringType (0..1)	The vesion id of the software of a module to check for.
hardware_major_release	oval-def:EntityStateVersionType (0..1)	The hardware major release of a module to check for.
hardware_individual_release	oval-def:EntityStateIntType (0..1)	The hardware individual release of a module to check for.
firmware_major_release	oval-def:EntityStateVersionType (0..1)	The major release of the firmware of a module to check for.
firmware_individual_release	oval-def:EntityStateIntType (0..1)	The individual release of the firmware of a module to check for.

< version55_test >

The version55_test is used to check the version of the Cisco CatOS operating system. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a version_object and the optional state element specifies the data to check.

The required information of version55_test can be got via a SHOW VERSION command. The separated version55_test enables an evaluation based on OVAL without having interactive command access to the device.

Extends: oval-def:TestType

Child Elements

Table 376: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< version55_object >

The version55_object element is used by a version55_test to define the different version information associated with a Cisco CatOS system. There is actually only one object relating to version and this is the system as a whole. Therefore, there are no child entities defined. Any OVAL Test written to check version will reference the same version5_object which is basically an empty object element.

Extends: oval-def:ObjectType

< version55_state >

The version55_state element defines the version information held within a Cisco CatOS software release. The switch_series element specifies the Catalyst switch series. The image_name element specifies the name of the CatOS image. The catos_release element specifies the software version information of the module.

Extends: oval-def:StateType

Child Elements

Table 377: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
switch_series	oval-def:EntityStateStringType (0..1)	The switch_series entity defines a target Catalyst switch series to check for. Each version of CatOS traditionally has target a specific Catalyst series of switches.
image_name	oval-def:EntityStateStringType (0..1)	The image_name entity defines a name of a CatOS image to check for.
catos_release	oval-def:EntityStateVersionType (0..1)	The catos_release entity defines a release version of CatOS to check for.

< version_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.5
- Reason: Replaced by the version55_test. Due to the fact it's not clear on how to separate the CatOS version, it was decided that the catos_major_release, catos_individual_release, and catos_version_id entities would be combined into a new single entity catos_release. A new test was created to reflect these changes. See the version55_test.
- Comment: This test has been deprecated and will be removed in version 6.0 of the language.

The version test is used to check the version of the Cisco CatOS operating system. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a version_object and the optional state element specifies the data to check.

The required information of version_test can be got via a SHOW VERSION command. The separated version_test enables an evaluation based on OVAL without having interactive command access to the device.

Extends: oval-def:TestType

Child Elements

Table 378: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< version_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.5
- Reason: Replaced by the version55_object. Due to the fact it's not clear on how to separate the CatOS version, it was decided that the catos_major_release, catos_individual_release, and catos_version_id entities would be combined into a new single entity catos_release. A new object was created to reflect these changes. See the version55_object.
- Comment: This object has been deprecated and will be removed in version 6.0 of the language.

The version_object element is used by a version test to define the different version information associated with a Cisco CatOS system. There is actually only one object relating to version and this is the system as a whole. Therefore, there are no child entities defined. Any OVAL Test written to check version will reference the same version_object which is basically an empty object element.

Extends: oval-def:ObjectType

< version_state > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.5
- Reason: Replaced by the version55_state. Due to the fact it's not clear on how to separate the CatOS version, it was decided that the catos_major_release, catos_individual_release, and catos_version_id entities would be combined into a new single entity catos_release. A new state was created to reflect these changes. See the version55_state.
- Comment: This state has been deprecated and will be removed in version 6.0 of the language.

The version_state element defines the version information held within a Cisco CatOS software release. The swtich_series element specifies the Catalyst switch series. The image_name element specifies the name of the CatOS image. The catos_major_release, catos_individual_release and catos_version_id elements specify the software version information of the module. For instance, if the CatOS version is 8.5(4c)GLX, then catos_major_release is 8.5GLX, catos_individual_release is 4 and catos_version_id is c.

Extends: oval-def:StateType

Child Elements

Table 379: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
switch_series	oval-def:EntityStateStringType (0..1)	A Catalyst switch series to check for.
image_name	oval-def:EntityStateStringType (0..1)	The name of a CatOS image to check for.
catos_major_release	oval-def:EntityStateVersionType (0..1)	The major release of CatOS to check for.
catos_individual_release	oval-def:EntityStateIntType (0..1)	The individual release of CatOS to check for.
catos_version_id	oval-def:EntityStateStringType (0..1)	The version id of Cat OS to check for.

Open Vulnerability and Assessment Language: CatOS System Characteristics

- Schema: CatOS System Characteristics
- Version: 5.11.1:1.1
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the Cisco CatOS specific system characteristic items found in Open Vulnerability and Assessment Language (OVAL). Each item is an extension of the standard test element defined in the Core Definition Schema. Through extension, each test inherits a set of elements and attributes that are shared amongst all OVAL tests. Each test is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core Definition Schema is not outlined here.

This schema was originally developed by Yuzheng Zhou at Hewlett-Packard. The OVAL Schema is maintained by the 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.cisecurity.org.

Item Listing

- *< line_item >*

- *< module_item >*
- *< version_item >*

< line_item >

Stores the properties of specific lines in the catos config file.

Extends: oval-sc:ItemType

Child Elements

Table 380: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
show_subcommand	oval-sc:EntityItemStringType (0..1)	The name of the SHOW sub-command.
config_line	oval-sc:EntityItemStringType (0..1)	The value returned from by the specified SHOW sub-command.

< module_item >

Stores results from SHOW MODULE command.

Extends: oval-sc:ItemType

Child Elements

Table 381: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
module_number	oval-sc:EntityItemIntType (0..1)	
type	oval-sc:EntityItemStringType (0..1)	
model	oval-sc:EntityItemStringType (0..1)	
software_major_release	oval-sc:EntityItemVersionType (0..1)	
software_individual_release	oval-sc:EntityItemIntType (0..1)	
software_version_id	oval-sc:EntityItemStringType (0..1)	
hardware_major_release	oval-sc:EntityItemVersionType (0..1)	
hardware_individual_release	oval-sc:EntityItemIntType (0..1)	
firmware_major_release	oval-sc:EntityItemVersionType (0..1)	
firmware_individual_release	oval-sc:EntityItemIntType (0..1)	

< version_item >

Stores results from SHOW VERSION command.

Extends: oval-sc:ItemType

Child Elements

Table 382: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
switch_series	oval-sc:EntityItemStringType (0..1)	The switch_series entity specifies the target Catalyst switch series for the given version of CatOS.
image_name	oval-sc:EntityItemStringType (0..1)	The image_name entity specifies the name of the CatOS image.
catos_release	oval-sc:EntityItemVersionType (0..1)	The catos_release entity specifies the release version of CatOS.
catos_major_release (Deprecated)	oval-sc:EntityItemVersionType (0..1)	
catos_individual_release (Deprecated)	oval-sc:EntityItemIntType (0..1)	
catos_version_id (Deprecated)	oval-sc:EntityItemStringType (0..1)	

Open Vulnerability and Assessment Language: IOS Definition

- Schema: IOS Definition
- Version: 5.11.1:1.2
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the IOS specific tests found in Open Vulnerability and Assessment Language (OVAL). Each test is an extension of the standard test element defined in the Core Definition Schema. Through extension, each test inherits a set of elements and attributes that are shared amongst all OVAL tests. Each test is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core Definition Schema is not outlined here.

The OVAL Schema is maintained by the 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.cisecurity.org.

Test Listing

- *< acl_test >*

- *< bgpneighbor_test >*
- *< global_test >*
- *< interface_test >*
- *< line_test >*
- *< router_test >*
- *< routingprotocolauthintf_test >*
- *< section_test >*
- *< snmp_test >*
- *< snmpcommunity_test >*
- *< snmpgroup_test >*
- *< snmphost_test >*
- *< snmpuser_test >*
- *< snmpview_test >*
- *< tclsh_test >*
- *< version55_test >*
- *< version_test >* (Deprecated)

< acl_test >

The acl test is used to check the properties of specific output lines from an ACL configuration.

Extends: oval-def:TestType

Child Elements

Table 383: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< acl_object >

The acl_object element is used by an acl test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An acl object consists of a an acl name and an IP version entity that is the name and the IP protocol version of the access-list to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 384: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityObjectStringType (1..1)	The name of the ACL.
ip_version	ios-def:EntityObjectAccessListIPVersionType (1..1)	The IP version of the ACL.
oval-def:filter	n/a (0..unbounded)	

< acl_state >

The acl_state element defines the different information that can be used to evaluate the result of a specific ACL configuration. This includes the name of ths ACL and the corresponding config lines. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 385: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	The name of the ACL.
ip_version	ios-def:EntityStateAccessListIPVersionType (0..1)	The IP version of the ACL.
use	ios-def:EntityStateAccessListUseType (0..1)	The feature where the ACL is used.
used_in	oval-def:EntityStateStringType (0..1)	The name of where the ACL is used. For example if use is 'INTERFACE', use_in will be the name of the interface.
interface_direction	ios-def:EntityStateAccessListInterfaceDirectionType (0..1)	The direction the ACL is applied on an interface.
acl_config_lines	oval-def:EntityStateStringType (0..1)	The value returned with all config lines of the ACL.
config_line	oval-def:EntityStateStringType (0..1)	The value returned with one ACL config line at a time.

< bgpneighbor_test >

The bgpneighbor test is used to check the bgp neighbpr properties of bgp instances instances in IOS.

Extends: oval-def:TestType

Child Elements

Table 386: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< bgpneighbor_object >

The bgpneighbor_object element is used by a bgpneighbor test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A bgpneighbor object consists of a neighbor entity.

Extends: oval-def:ObjectType

Child Elements

Table 387: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
neighbor	oval-def:EntityObjectStringType (1..1)	The bgp neighbor.
oval-def:filter	n/a (0..unbounded)	

< bgpneighbor_state >

The bgpneighbor_state element defines the different information that can be used to evaluate the result of a bgp neighbor configuration. This includes the neighbor and the password option, if configured. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 388: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
neighbor	oval-def:EntityStateStringType (0..1)	The bgp neighbor.
password	oval-def:EntityStateStringType (0..1)	The bgp authentication password, if configured. If Encryption type is configured it should be included in the password string. For example '0 cisco123'.

< global_test >

The global test is used to check for the existence of a particular line in the ios config file under the global context. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a global_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 389: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< global_object >

The global_object element is used by a global test to define the object to be evaluated. For the most part this object checks for existence and is used without a state comparision. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 390: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
global_command	oval-def:EntityObjectStringType (1..1)	The global_command entity identifies a specific line in the ios config file under the global context.
oval-def:filter	n/a (0..unbounded)	

< global_state >

The global_state element defines the different information that can be found in the ios config file under the global context. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 391: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
global_command	oval-def:EntityStateStringType (0..1)	The global_command entity identifies a specific line in the ios config file under the global context.

< interface_test >

The interface test is used to check for the existence of a particular interface on the Cisco IOS device. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a interface_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 392: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< interface_object >

The interface_object element is used by an interface_test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An interface_object consists of a name entity that is the name of the IOS interface to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 393: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityObjectStringType (1..1)	
oval-def:filter	n/a (0..unbounded)	

< interface_state >

The interface_state element defines the different information that can be used to evaluate the result of a specific IOS interface. This includes the name, status, and address information about the interface. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 394: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	
ip_directed_broadcast_command	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	Directed broadcast command enabled on the interface. The default is false.
no_ip_directed_broadcast_command (Deprecated)	oval-def:EntityStateStringType (0..1)	
proxy_arp_command	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	Element that is true if the proxy_arp command is enabled on the interface. The default is true.
shutdown_command	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	Element that is true if the interface is shut down. The default is false.
hardware_addr	oval-def:EntityStateStringType (0..1)	The interface hardware (MAC) address.
ipv4_address	oval-def:EntityStateIPAddressStringType (0..1)	The interface IPv4 address and mask. This element should only allow 'ipv4_address' of the oval:SimpleDatatypeEnumeration.
ipv6_address	oval-def:EntityStateIPAddressStringType (0..1)	The interface IPv6 address and mask. This element should only allow 'ipv6_address' of the oval:SimpleDatatypeEnumeration.
ipv4_access_list	oval-def:EntityStateStringType (0..1)	The ingress or egress IPv4 ACL name applied on the interface.
ipv6_access_list	oval-def:EntityStateStringType (0..1)	The ingress or egress IPv6 ACL name applied on the interface.
crypto_map	oval-def:EntityStateStringType (0..1)	The crypto map name applied to the interface.
ipv4_urpf_command	oval-def:EntityStateStringType (0..1)	The IPv4 uRPF command under the interface.
ipv6_urpf_command	oval-def:EntityStateStringType (0..1)	The IPv6 uRPF command under the interface.
urpf_command (Deprecated)	oval-def:EntityStateStringType (0..1)	The uRPF command under the interface.
switchport_trunk_encapsulation	ios-def:EntityStateTrunkEncapType (0..1)	The switchport trunk encapsulation option configured on the interface (if applicable).
switchport_mode	ios-def:EntityStateSwitchportModeType (0..1)	The switchport mode option configured on the interface (if applicable).
switchport_native_vlan	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	The trunk native vlan configured on the interface (if applicable).
switchport_access_vlan	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	The access vlan configured on the interface (if applicable).
switchport_trunked_vlans	oval-def:EntityStateStringType (0..1)	The vlans that are trunked configured on the interface (if applicable).
switchport_pruned_vlans	oval-def:EntityStateStringType (0..1)	The vlans that are pruned from the trunk (if applicable).
switchport_port_security	oval-def:EntityStateStringType (0..1)	The switchport port-security commands configured on the interface (if applicable).

< line_test >

The line test is used to check the properties of specific output lines from a SHOW command, such as show running-config. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a line_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 395: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< line_object >

The line_object element is used by a line test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A line object consists of a show_subcommand entity that is the name of a SHOW sub-command to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 396: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
show_subcommand	oval-def:EntityObjectStringType (1..1)	The name of a SHOW sub-command.
oval-def:filter	n/a (0..unbounded)	

< line_state >

The line_state element defines the different information that can be used to evaluate the result of a specific SHOW sub-command. This includes the name of ths sub-command and the corresponding config line. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 397: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
show_subcommand	oval-def:EntityStateStringType (0..1)	The name of the SHOW sub-command.
config_line	oval-def:EntityStateStringType (0..1)	The value returned from by the specified SHOW sub-command.

< router_test >

The router test is used to check the properties of specific output lines from a router configurated instance in IOS.

Extends: oval-def:TestType

Child Elements

Table 398: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< router_object >

The router_object element is used by a router test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A router object consists of a router protocol and router identifier entity.

Extends: oval-def:ObjectType

Child Elements

Table 399: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
protocol	ios-def:EntityObjectRoutingProtocolType (1..1)	The routing protocol of the router instance.
id	oval-def:EntityObjectIntType (1..1)	The IOS router id.
oval-def:filter	n/a (0..unbounded)	

< router_state >

The router_state element defines the different information that can be used to evaluate the result of a specific router command. This includes the protocol of the router instance, the id, the networks, bgp neighbor, ospf authentication area commands and the corresponding config lines. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 400: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
protocol	ios-def:EntityStateRoutingProtocolType (1..1)	The routing protocol of the router instance. If there are more than one router configurations, for example ospf instances, different objects should be created for each.
id	oval-def:EntityStateIntType (0..1)	The IOS router id
network	oval-def:EntityStateStringType (0..1)	The subnet in the network command of the router instance. The area can be included in the string for OSPF.
bgp_neighbor	oval-def:EntityStateStringType (0..1)	The BGP neighbors, if applicable.
ospf_authentication_area	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	The OSPF area that is authenticated, if applicable.
router_config_lines	oval-def:EntityStateStringType (0..1)	The value returned with all config lines of the router instance.

< routingprotocolauthintf_test >

The routing protocol authentication interface test is used to check the properties of routing protocol authentication configured under interfaces in IOS.

Extends: oval-def:TestType

Child Elements

Table 401: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< routingprotocolauthintf_object >

The routingprotocolauthintf_object element is used by a routingprotocolauthintf test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A routingprotocolauthintf object consists of an interface and the routing protocol that is authenticated entity.

Extends: oval-def:ObjectType

Child Elements

Table 402: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
interface	oval-def:EntityObjectStringType (1..1)	The interface name.
protocol	ios-def:EntityObjectRoutingProtocolType (1..1)	The routing protocol.
oval-def:filter	n/a (0..unbounded)	

< routingprotocolauthintf_state >

The routingprotocolauthintf_state element defines the different information that can be used to evaluate the result of a specific routing protocol interface authentication configurations. This includes the interface, the protocol, the id, the authentication type, the ospf area, the key chain command and the corresponding config lines. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 403: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
interface	oval-def:EntityStateStringType (0..1)	The interface name.
protocol	ios-def:EntityStateRoutingProtocolType (0..1)	The routing protocol.
id	oval-def:EntityStateIntType (0..1)	The routing protocol id, if applicable.
auth_type	ios-def:EntityStateRoutingAuthTypeStringType (0..1)	The routing protocol authentication type.
ospf_area	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	The OSPF area that is authenticated, if applicable.
key_chain	oval-def:EntityStateStringType (0..1)	The name of the key chain, if applicable.

< section_test >

The section test is used to check the properties of specific output lines from a configuration section.

Extends: oval-def:TestType

Child Elements

Table 404: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< section_object >

The section_object element is used by a section test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A section object consists of a section_command entity that is the name of a section command to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 405: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
section_command	oval-def:EntityObjectStringType (1..1)	The name of a section command.
oval-def:filter	n/a (0..unbounded)	

< section_state >

The section_state element defines the different information that can be used to evaluate the result of a specific section command. This includes the name of ths section_command and the corresponding config lines. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 406: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
section_command	oval-def:EntityStateStringType (0..1)	The name of the section command.
section_config_lines	oval-def:EntityStateStringType (0..1)	The value returned with all config lines of the section.
config_line	oval-def:EntityStateStringType (0..1)	The value returned with one config line of the section at a time.

< snmp_test >

Tests if lines under the global context associated with snmp that have a specifiec access list or community name.

Extends: oval-def:TestType

Child Elements

Table 407: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< snmp_object >

The snmp_object element is used by a snmp test to define those objects to evaluated based on a specified state. There is actually only one object relating to snmp and this is the system as a whole. Therefore, there are no child entities defined. Any OVAL Test written to check snmp will reference the same snmp_object which is basically an empty object element.

Extends: oval-def:ObjectType

< snmp_state >

Extends: oval-def:StateType

Child Elements

Table 408: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
access_list	oval-def:EntityStateStringType (0..1)	
community_name	oval-def:EntityStateStringType (0..1)	

< snmpcommunity_test >

The snmpcommunity test is used to check the properties of specific output lines from an SNMP configuration.

Extends: oval-def:TestType

Child Elements

Table 409: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< snmpcommunity_object >

The snmpcommunity_object element is used by an snmpcommunity test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An snmpcommunity object consists of a community name entity to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 410: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityObjectStringType (1..1)	The SNMP community name.
oval-def:filter	n/a (0..unbounded)	

< snmpcommunity_state >

The snmpcommunity_state element defines the different information that can be used to evaluate the result of a specific ‘snmp community’ IOS command. This includes the community name and the corresponding options. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 411: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	The SNMP community name.
view	oval-def:EntityStateStringType (0..1)	The view that restricts the OIDs of this community.
mode	ios-def:EntityStateSNMPModeStringType (0..1)	The read-write privileges of the community.
ipv4_acl	oval-def:EntityStateStringType (0..1)	The IPv4 ACL name applied to the community.
ipv6_acl	oval-def:EntityStateStringType (0..1)	The IPv6 ACL name applied to the community.

< snmpgroup_test >

The snmpgroup test is used to check the properties of specific output lines from an SNMP group configuration.

Extends: oval-def:TestType

Child Elements

Table 412: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< snmpgroup_object >

The snmpgroup_object element is used by an snmpgroup test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A snmpgroup object consists of a name entity that is the name of the SNMP group to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 413: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityObjectStringType (1..1)	The SNMP group name.
oval-def:filter	n/a (0..unbounded)	

< snmpgroup_state >

The snmpgroup_state element defines the different information that can be used to evaluate the result of a specific 'snmp-server group' IOS command. This includes the user name and the corresponding options. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 414: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	The SNMP group name.
version	ios-def:EntityStateSNMPVersionStringType (0..1)	The SNMP version of the group.
snmpv3_sec_level	ios-def:EntityStateSNMPSecLevelStringType (0..1)	The SNMPv3 security configured for the group.
ipv4_acl	oval-def:EntityStateStringType (0..1)	The IPv4 ACL name applied to the group.
ipv6_acl	oval-def:EntityStateStringType (0..1)	The IPv6 ACL name applied to the group.
read_view	oval-def:EntityStateStringType (0..1)	The SNMP read view applied to the group.
write_view	oval-def:EntityStateStringType (0..1)	The SNMP write view applied to the group.
notify_view	oval-def:EntityStateStringType (0..1)	The SNMP notify view applied to the group.

< snmphost_test >

The snmphost test is used to check the properties of specific output lines from an SNMP configuration.

Extends: oval-def:TestType

Child Elements

Table 415: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< snmphost_object >

The snmphost_object element is used by an snmphost test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A snmphost object consists of a host entity that is the host of the 'snmp host' IOS command to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 416: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
host	oval-def:EntityObjectStringType (1..1)	The SNMP host address or hostname.
oval-def:filter	n/a (0..unbounded)	

< snmphost_state >

The snmphost_state element defines the different information that can be used to evaluate the result of a specific 'snmp host' IOS command. This includes the host and the corresponding options. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 417: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
host	oval-def:EntityStateStringType (0..1)	The SNMP host address or hostname.
community_or_user	oval-def:EntityStateStringType (0..1)	The community string or SNMPv3 user configured for the host.
version	ios-def:EntityStateSNMPVersionStringType (0..1)	The SNMP version.
snmpv3_sec_level	ios-def:EntityStateSNMPSecLevelStringType (0..1)	The SNMPv3 security configured for the host.
traps	oval-def:EntityStateStringType (0..1)	The SNMP traps configured.

< snmpuser_test >

The snmpuser test is used to check the properties of specific output lines from an SNMP user configuration.

Extends: oval-def:TestType

Child Elements

Table 418: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< snmpuser_object >

The snmpuser_object element is used by an snmpuser test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A snmpuser object consists of a name entity that is the name of the SNMP user to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 419: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityObjectStringType (1..1)	The SNMP user name.
oval-def:filter	n/a (0..unbounded)	

< snmpuser_state >

The snmpuser_state element defines the different information that can be used to evaluate the result of a specific ‘show snmp user’ IOS command. This includes the user name and the corresponding options. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 420: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	The SNMP user name.
group	oval-def:EntityStateStringType (0..1)	The SNMP group the user belongs to.
version	ios-def:EntityStateSNMPVersionStringType (0..1)	The SNMP version of the user.
ipv4_acl	oval-def:EntityStateStringType (0..1)	The IPv4 ACL name applied to the user.
ipv6_acl	oval-def:EntityStateStringType (0..1)	The IPv6 ACL name applied to the user.
priv	ios-def:EntityStateSNMPPrivStringType (0..1)	The SNMP encryption type for the user (for SNMPv3).
auth	ios-def:EntityStateSNMPAuthStringType (0..1)	The SNMP authentication type for the user (for SNMPv3).

< snmpview_test >

The snmpview test is used to check the properties of specific output lines from an SNMP view configuration.

Extends: oval-def:TestType

Child Elements

Table 421: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< snmpview_object >

The snmpview_object element is used by an snmpview test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A snmpview object consists of a name entity that is the name of the SNMP view to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 422: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityObjectStringType (1..1)	The SNMP view name.
oval-def:filter	n/a (0..unbounded)	

< snmpview_state >

The snmpview_state element defines the different information that can be used to evaluate the result of a specific 'snmp-server view' IOS command. This includes the view name and the corresponding options. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 423: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	The SNMP view name.
mib_family	oval-def:EntityStateStringType (0..1)	The SNMP MIB family of the view.
include	oval-def:EntityStateBoolType (0..1)	It is true if the included option is used in the view.

< tclsh_test >

The tclsh test is used to check tclsh information of the IOS operating system. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a tclsh_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 424: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< tclsh_object >

The tclsh_object element is used by a tclsh test to define those objects to evaluated based on a specified state. There is actually only one object relating to tchlsh and this is the system as a whole. Therefore, there are no child entities defined. Any OVAL Test written to check tclsh will reference the same tclsh_object which is basically an empty object element.

Extends: oval-def:ObjectType

< tclsh_state >

The tclsh_state element defines information about TCLSH. This includes the available entity which describes whether TCLSH is available on the system. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 425: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
available	oval-def:EntityStateBoolType (0..1)	This boolean entity describes whether TCLSH is available on the system. A value of true means that TCLSH is available.

< version55_test >

The version55_test is used to check the version of the IOS operating system. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a version_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 426: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< version55_object >

The version55_object element is used by a version55_test to define the different version information associated with an IOS system. There is actually only one object relating to version and this is the system as a whole. Therefore, there are no child entities defined. Any OVAL Test written to check version will reference the same version55_object which is basically an empty object element.

Extends: oval-def:ObjectType

< version55_state >

The version55_state element defines the version information held within a Cisco IOS Train. A Cisco IOS train is a vehicle for delivering releases that evolve from a common code base.

Extends: oval-def:StateType

Child Elements

Table 427: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
major_version	oval-def:EntityStateIntType (0..1)	The major_version entity is used to check the major version piece of the version string. The value is an integer and in the example 12.4(9)T0a the major version is '12'.
minor_version	oval-def:EntityStateIntType (0..1)	The minor_version entity is used to check the minor version piece of the version string. The value is an integer and in the example 12.4(9)T0a the minor version is '4'.
release	oval-def:EntityStateIntType (0..1)	The release entity is used to check the release piece of the version string. The value is an integer and in the example 12.4(9)T0a the release is '9'.
train_identifier	oval-def:EntityStateStringType (0..1)	The train_identifier entity is used to check the type of train represented in the version string. The value is a string and in the example 12.4(9)T0a the train identifier is 'T'. The following explaination from Wikipedia should help explain the different train identifiers. Cisco IOS releases are split into several "trains", each containing a different set of features. Trains more or less map onto distinct markets or groups of customers that Cisco is targeting. The 'mainline' train is designed to be the most stable release the company can offer, and its feature set never expands during its lifetime. Updates are released only to address bugs in the product. The previous technology train becomes the source for the current mainline train–for example, the 12.1T train becomes the basis for the 12.2 mainline. Therefore, to determine the features available in a particular mainline release, look at the previous T train release. The 'T' (Technology) train, gets new features and bug fixes throughout its life, and is therefore less stable than the mainline. (In releases prior to Cisco IOS Release 12.0, the P train served as the Technology train.) The 'S' (Service Provider) train, runs only on the company's core router products and is heavily customized for Service Provider customers. The 'E' (Enterprise) train, is customized for implementation in enterprise environments. The 'B' (broadband) train, support internet based broadband features. The 'XA', 'Xb' … (special functionality) train, needs to be documented. There are other trains from time to time, designed for specific needs – for example, the 12.0AA train contained new code required for Cisco's AS5800 product.
rebuild	oval-def:EntityStateIntType (0..1)	The rebuild entity is used to check the rebuild piece of the version string. The value is an integer and in the example 12.4(9)T0a the rebuild is '0'. Often a rebuild is compiled to fix a single specific problem or vulnerability for a given IOS version. For example, 12.1(8)E14 is a Rebuild, the 14 denoting the 14th rebuild of 12.1(8)E. Rebuilds are produced to either quickly repair a defect, or to satisfy customers who do not want to upgrade to a later major revision because they may be running critical infrastructure on their devices, and hence prefer to minimise change and risk.
subrebuild	oval-def:EntityStateStringType (0..1)	The subrebuild entity is used to check the subrebuild piece of the version string. The value is a string and in the example 12.4(9)T0a the subrebuild is 'a'.
mainline_rebuild	oval-def:EntityStateStringType (0..1)	The mainline_rebuild entity is used to check the mainline rebuild piece of the version string. The mainline rebuild is just a regular rebuild release against the mainline operating system release (e.g. the branch of development that would typically be called "the trunk" that isn't associated with a train). Since there is no train identifier to stick the rebuild release after, they stick a alphabetic character inside the parens holding the maintenance release number. For example, 12.4(5b) is the second rebuild of the 12.4(5) maintenance release.
version_string	oval-def:EntityStateIOSVersionType (0..1)	The version_string entity is used to check the raw string output of a 'show version' command.

< version_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.5
- Reason: Replaced by the version55_test. Additional IOS version components were added to the version_state in order to support a wider range of IOS version strings. Also, the major_release and train_number entities were removed from the version_state element. A new test was created to reflect these changes. See the version55_test.
- Comment: This test has been deprecated and will be removed in version 6.0 of the language.

The version test is used to check the version of the IOS operating system. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a version_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 428: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< version_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.5
- Reason: Replaced by the version55_object. Additional IOS version components were added to the version_state in order to support a wider range of IOS version strings. Also, the major_release and train_number entities were removed from the version_state element. A new object was created to reflect these changes. See the version55_object.
- Comment: This object has been deprecated and will be removed in version 6.0 of the language.

The version_object element is used by a version test to define the different version information associated with an IOS system. There is actually only one object relating to version and this is the system as a whole. Therefore, there are no child entities defined. Any OVAL Test written to check version will reference the same version_object which is basically an empty object element.

Extends: oval-def:ObjectType

< version_state > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.5

- Reason: Replaced by the version55_state. Additional IOS version components were added to the version_state in order to support a wider range of IOS version strings. Also, the major_release and train_number entities were removed from this version_state element. A new state was created to reflect these changes. See the version55_state.
- Comment: This state has been deprecated and will be removed in version 6.0 of the language.

The version_state element defines the version information held within a Cisco IOS Train. A Cisco IOS train is a vehicle for delivering releases that evolve from a common code base.

Extends: oval-def:StateType

Child Elements

Table 429: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
major_release	oval-def:EntityStateStringType (0..1)	The major_release is a combination of train and rebuild information and is used by Cisco advisories to identify major releases.
train_number	oval-def:EntityStateStringType (0..1)	The train number is the dotted version that starts a version string. For example the version string 12.2(3)T has a train number of 12.2.
train_identifier	ios-def:EntityStateTrainIdentifierType (0..1)	The train identifier is the type of Train. For example the version string 12.2(3)T has a train identifier of T. Please see the EntityStateVersionTrainIdentifierType for more information about the different train identifiers.
version_string	oval-def:EntityStateIOSVersionType (0..1)	The version is the raw string output of a 'show version' command.

== EntityObjectAccessListIPVersionType ==

The EntityObjectAccessListIPVersionType complex type restricts a string value to a specific set of values: IPV4, IPV6. These values describe if an ACL is for IPv4 or IPv6 in a Cisco IOS configuration. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityObjectStringType

Table 430: Enumeration Values

Value	Description
IPV4	(No Description)
IPV6	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityObjectRoutingProtocolType ==

The EntityObjectRoutingProtocolType complex type restricts a string value to a specific set of values: EIGRP, OSPF, BGP, RIP, RIPV2, ISIS. These values describe the routing protocol used in a Cisco IOS configuration. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityObjectStringType

Table 431: Enumeration Values

Value	Description
EIGRP	(No Description)
OSPF	(No Description)
BGP	(No Description)
RIP	(No Description)
RIPV2	(No Description)
ISIS	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateAccessListInterfaceDirectionType ==

The EntityStateAccessListInterfaceDirectionType complex type restricts a string value to a specific set of values: IN, OUT. These values describe the inbound or outbound ACL direction on an interface in a Cisco IOS configuration. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 432: Enumeration Values

Value	Description
IN	(No Description)
OUT	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateAccessListIPVersionType ==

The EntityStateRoutingProtocolType complex type restricts a string value to a specific set of values: IPV4, IPV6. These values describe if an ACL is for IPv4 or IPv6 in a Cisco IOS configuration. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 433: Enumeration Values

Value	Description
IPV4	(No Description)
IPV6	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateAccessListUseType ==

The EntityStateAccessListUseType complex type restricts a string value to a specific set of values: INTERFACE, CRYPTO_MAP_MATCH, CLASS_MAP_MATCH, ROUTE_MAP_MATCH, IGMP_FILTER, VTY. These values describe the ACL use in a Cisco IOS configuration. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 434: Enumeration Values

Value	Description
INTERFACE	(No Description)
CRYPTO_MAP_MATCH	(No Description)
CLASS_MAP_MATCH	(No Description)
ROUTE_MAP_MATCH	(No Description)
IGMP_FILTER	(No Description)
VTY	(No Description)
NONE (Deprecated)	**Deprecated As Of Version:** 5.11.2:1.0 **Reason:** The EntityStateSimpleBaseType check_existence attribute serves the same purpose as this enumeration value. **Comment:** This AccessListUseType enumeration value has been deprecated and may be removed in a future version of the language.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateRoutingAuthTypeStringType ==

The EntityStateRoutingAuthTypeStringType complex type restricts a string value to a specific set of values: CLEARTEXT, MESSAGE_DIGEST. These values describe the routing protocol authentication types used in a Cisco IOS configuration. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 435: Enumeration Values

Value	Description
CLEARTEXT	(No Description)
MESSAGE_DIGEST	(No Description)
NULL (Deprecated)	**Deprecated As Of Version:** 5.11.2:1.0 **Reason:** The NULL authentication area type is never declared in an interface ip ospf command context. **Comment:** This RoutingAuthTypeStringType enumeration value has been deprecated and may be removed in a future version of the language.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateRoutingProtocolType ==

The EntityStateRoutingProtocolType complex type restricts a string value to a specific set of values: EIGRP, OSPF, BGP, RIP, RIPV2, ISIS. These values describe the routing protocol used in a Cisco IOS configuration. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 436: Enumeration Values

Value	Description
EIGRP	(No Description)
OSPF	(No Description)
BGP	(No Description)
RIP	(No Description)
RIPV2	(No Description)
ISIS	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateSNMPVersionStringType ==

The EntityStateSNMPVersionStringType complex type restricts a string value to a specific set of values: 1, 2c, 3. These values describe the SNMP version in a Cisco IOS configuration. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 437: Enumeration Values

Value	Description
1	(No Description)
2C	(No Description)
3	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateSNMPSecLevelStringType ==

The EntityStateSNMPVersionStringType complex type restricts a string value to a specific set of values: PRIV, AUTH, NO_AUTH. These values describe the SNMP security level (encryption, Authentication, None) in a Cisco IOS SNMPv3 related configurations. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 438: Enumeration Values

Value	Description
PRIV	(No Description)
AUTH	(No Description)
NO_AUTH	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateSNMPModeStringType ==

The EntityStateSNMPModeStringType complex type restricts a string value to a specific set of values: RO, RW. These values describe the SNMP mode (read-only, read-write) in a Cisco IOS SNMPv3 related configurations. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 439: Enumeration Values

Value	Description
RO	(No Description)
RW	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateSNMPAuthStringType ==

The EntityStateSNMPAuthStringType complex type restricts a string value to a specific set of values: MD5, SHA. These values describe the authentication algorithm in a Cisco IOS SNMPv3 related configurations. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 440: Enumeration Values

Value	Description
MD5	(No Description)
SHA	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateSNMPPrivStringType ==

The EntityStateSNMPPrivStringType complex type restricts a string value to a specific set of values: DES, 3DES, AES. These values describe the encryption algorithm in a Cisco IOS SNMPv3 related configurations. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 441: Enumeration Values

Value	Description
DES	(No Description)
3DES	(No Description)
AES	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateSwitchportModeType ==

The EntityObjectRoutingProtocolType complex type restricts a string value to a specific set of values: DYNAMIC, TRUNK, ACCESS. These values describe the interface switchport mode types in IOS. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 442: Enumeration Values

Value	Description
DYNAMIC	(No Description)
TRUNK	(No Description)
ACCESS	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateTrainIdentifierType == (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.5
- Reason: Additional IOS version components were added to the version_state in order to support a wider range of IOS version strings. Also, the train_number entity, which uses this enumeration, was removed from the version_state element. As a result, this enumeration is no longer needed.
- Comment: This enumeration has been deprecated and will be removed in version 6.0 of the language.

The EntityStateTrainIdentifierType complex type restricts a string value to a specific set of values. These values describe the possible types of trains in a Cisco IOS release. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 443: Enumeration Values

Value	Description
mainline	The mainline Train consolidates releases and fixes defects. Inherits features from the parent T train, and does not add additional features.
T	Introduces new features and fixes defects.
S	Consolidates 12.1E, 12.2 mainline, and 12.0S, which supports high-end backbone routing, and fixes defects.
E	Targets enterprise core and SP edge, supports advanced QoS, voice, security, and firewall, and fixes defects.
B	Supports broadband features and fixes defects.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateTrunkEncapType ==

The EntityStateTrunkEncapType complex type restricts a string value to a specific set of values: DOT1Q, ISL, NEGOTIATE. These values describe the interface trunk encapsulation types on an interfaces in IOS. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 444: Enumeration Values

Value	Description
DOT1Q	(No Description)
ISL	(No Description)
NEGOTIATE	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

Open Vulnerability and Assessment Language: IOS Definition

- Schema: IOS Definition
- Version: 5.11.1:1.2
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the IOS specific system characteristic items found in Open Vulnerability and Assessment Language (OVAL). Each item is an extension of the standard item element defined in the Core System Characteristic Schema. Through extension, each item inherits a set of elements and attributes that are shared amongst all OVAL Items. Each item is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core System Characteristic Schema is not outlined here.

The OVAL Schema is maintained by the 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.cisecurity.org.

Item Listing

- *< acl_item >*
- *< bgpneighbor_item >*
- *< global_item >*
- *< interface_item >*
- *< line_item >*
- *< router_item >*
- *< routingprotocolauthintf_item >*
- *< section_item >*
- *< snmp_item >*
- *< snmpcommunity_item >*
- *< snmpgroup_item >*
- *< snmphost_item >*
- *< snmpuser_item >*
- *< snmpview_item >*
- *< tclsh_item >*
- *< version_item >*

< acl_item >

Stores command that are part of a IOS configuration section. For example all configuration lines under an interface. It should not store configurations for configs that already have a separate item. For example BGP has a router item and should not also be stored in a acl_item.

Extends: oval-sc:ItemType

Child Elements

Table 445: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	Element with the name of the ACL.
ip_version	ios-sc:EntityItemAccessListIPVersionType (0..1)	Element with the IP version of the ACL.
use	ios-sc:EntityItemAccessListUseType (0..1)	Element with the feature where the ACL is used. If the same ACL is applied in more than one feature (i.e interface and crypto map), multiple items needs to be created.
used_in	oval-sc:EntityItemStringType (0..1)	Element with the name of where the ACL is used. For example if use is 'INTERFACE', use_in will be the name of the interface. If the same ACL is applied in more than one feature (i.e interface and crypto map), multiple items needs to be created.
interface_direction	ios-sc:EntityItemAccessListInterfaceDirectionType (0..1)	Element with the direction the ACL is applied on an interface.
acl_config_lines	oval-sc:EntityItemStringType (0..1)	Element with the value returned with all config lines of the ACL.
config_line	oval-sc:EntityItemStringType (0..unbounded)	Element with the value returned with one ACL config line at a time.

< bgpneighbor_item >

Stores information about bgp neighbors configured in bgp instances.

Extends: oval-sc:ItemType

Child Elements

Table 446: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
neighbor	oval-sc:EntityItemStringType (0..1)	Element with the bgp neighbor.
password	oval-sc:EntityItemStringType (0..1)	Element with the bgp authentication password, if configured. If Encryption type is configured it should be included in the password string. For example '0 cisco123'.

< global_item >

Sotres information about the existence of a particular line in the ios config file under the global context.

Extends: oval-sc:ItemType

Child Elements

Table 447: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
global_command	oval-sc:EntityItemStringType (0..1)	

< interface_item >

The interface_item represents an IOS interface and its configuration options.

Extends: oval-sc:ItemType

Child Elements

Table 448: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	Element with the interface name.
ip_directed_broadcast_command	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	Element that is true if the directed broadcast command is enabled on the interface. The default is false.
no_ip_directed_broadcast_command (Deprecated)	oval-sc:EntityItemStringType (0..1)	
proxy_arp_command	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	Element that is true if the proxy_arp command is enabled on the interface. The default is true.
shutdown_command	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	Element that is true if the interface is shut down. The default is false.
hardware_addr	oval-sc:EntityItemStringType (0..1)	Element with the interface hardware (MAC) address.
ipv4_address	oval-sc:EntityItemIPAddressStringType (0..1)	Element with the interface IPv4 address and mask. This element should only allow 'ipv4_address' of the oval:SimpleDatatypeEnumeration.
ipv6_address	oval-sc:EntityItemIPAddressStringType (0..unbounded)	Element with the interface IPv6 address and mask. This element should only allow 'ipv6_address' of the oval:SimpleDatatypeEnumeration.
ipv4_access_list	oval-sc:EntityItemStringType (0..2)	Element with the ingress or egress IPv4 ACL name applied on the interface.
ipv6_access_list	oval-sc:EntityItemStringType (0..2)	Element with the ingress or egress IPv6 ACL name applied on the interface.
crypto_map	oval-sc:EntityItemStringType (0..1)	Element with the crypto map name applied to the interface.
ipv4_urpf_command	oval-sc:EntityItemStringType (0..1)	Element with the uRPF command for IPv4 under the interface.
ipv6_urpf_command	oval-sc:EntityItemStringType (0..1)	Element with the uRPF command for IPv6 under the interface.
urpf_command (Deprecated)	oval-sc:EntityItemStringType (0..1)	Element with the uRPF command under the interface.
switchport_trunk_encapsulation	ios-sc:EntityItemTrunkEncapType (0..1)	Element with the switchport trunk encapsulation option configured on the interface (if applicable).
switchport_mode	ios-sc:EntityItemSwitchportModeType (0..1)	Element with the switchport mode option configured on the interface (if applicable).
switchport_native_vlan	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	Element with the trunk native vlan configured on the interface (if applicable).
switchport_access_vlan	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	Element with the access vlan configured on the interface (if applicable).
switchport_trunked_vlans	oval-sc:EntityItemStringType (0..1)	Element with the vlans that are trunked configured on the interface (if applicable).
switchport_pruned_vlans	oval-sc:EntityItemStringType (0..1)	Element with the vlans that are pruned from the trunk (if applicable).
switchport_port_security	oval-sc:EntityItemStringType (0..1)	Element with the switchport port-security commands configured on the interface (if applicable).

< line_item >

Stores the properties of specific lines in the ios config file.

Extends: oval-sc:ItemType

Child Elements

Table 449: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
show_subcommand	oval-sc:EntityItemStringType (0..1)	The name of the SHOW sub-command.
config_line	oval-sc:EntityItemStringType (0..1)	The value returned from by the specified SHOW sub-command.

< router_item >

Stores commands that are part of a IOS ‘router’ command configuration. For example ‘router bgp 123’.

Extends: oval-sc:ItemType

Child Elements

Table 450: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
protocol	ios-sc:EntityItemRoutingProtocolType (0..1)	Element with the routing protocol.
id	oval-sc:EntityItemIntType (0..1)	Element with the IOS router id.
network	oval-sc:EntityItemStringType (0..unbounded)	Element with the subnet in the network command of the router instance. The area can be included in the string for OSPF.
bgp_neighbor	oval-sc:EntityItemStringType (0..unbounded)	Element with the BGP neighbors, if applicable.
ospf_authentication_area	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..unbounded)	Element with the OSPF area that is authenticated, if applicable.
router_config_lines	oval-sc:EntityItemStringType (0..1)	Element with all config lines of the router.

< routingprotocolauthintf_item >

Stores information for routing protocol authentication configured under specific interfaces.

Extends: oval-sc:ItemType

Child Elements

Table 451: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
interface	oval-sc:EntityItemStringType (0..1)	Element with the interface.
protocol	ios-sc:EntityItemRoutingProtocolType (0..1)	Element with the routing protocol.
id	oval-sc:EntityItemIntType (0..1)	Element with the routing protocol id.
auth_type	ios-sc:EntityItemRoutingAuthTypeStringType (0..1)	Element with the routing protocol authentication type.
ospf_area	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	Element with the OSPF area that is authenticated, if applicable.
key_chain	oval-sc:EntityItemStringType (0..1)	Element with the name of the key chain, if applicable.

< section_item >

Stores command that are part of a IOS configuration section. For example all configuration lines under an interface. It should not store configurations for configs that already have a separate item. For example BGP has a router item and should not also be stored in a section_item.

Extends: oval-sc:ItemType

Child Elements

Table 452: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
section_command	oval-sc:EntityItemStringType (0..1)	The name of the section command.
section_config_lines	oval-sc:EntityItemStringType (0..1)	Element with all config lines of the section.
config_line	oval-sc:EntityItemStringType (0..unbounded)	Element with one config line of the section at a time.

< snmp_item >

Stores results from collecting lines under the global context associated with snmp.

Extends: oval-sc:ItemType

Child Elements

Table 453: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
access_list	oval-sc:EntityItemStringType (0..1)	
community_name	oval-sc:EntityItemStringType (0..1)	

< snmpcommunity_item >

Stores information about an SNMP community configuration in IOS. That information includes the community name, the view (if it applies) name, the read-write mode and the ACLs names applied.

Extends: oval-sc:ItemType

Child Elements

Table 454: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	Element with the SNMP community name.
view	oval-sc:EntityItemStringType (0..1)	Element with the view that restricts the OIDs of this community.
mode	ios-sc:EntityItemSNMPModeStringType (0..1)	Element with the read-write privileges of the community.
ipv4_acl	oval-sc:EntityItemStringType (0..1)	Element with the IPv4 ACL name applied to the community.
ipv6_acl	oval-sc:EntityItemStringType (0..1)	Element with the IPv6 ACL name applied to the community.

< snmpgroup_item >

Stores information about an SNMP group configuration in IOS. That information includes the group name, the SNMP version, the IPv4 or IPv6 ACL it is applied toand the read, write and/or notify views applied to the group.

Extends: oval-sc:ItemType

Child Elements

Table 455: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	Element with the SNMP group name.
version	ios-sc:EntityItemSNMPVersionStringType (0..1)	Element with the SNMP version of the group.
snmpv3_sec_level	ios-sc:EntityItemSNMPSecLevelStringType (0..1)	Element with the SNMPv3 security configure for the group.
ipv4_acl	oval-sc:EntityItemStringType (0..1)	Element with the IPv4 ACL name applied to the group.
ipv6_acl	oval-sc:EntityItemStringType (0..1)	Element with the IPv6 ACL name applied to the group.
read_view	oval-sc:EntityItemStringType (0..1)	Element with the SNMP read view applied to the group.
write_view	oval-sc:EntityItemStringType (0..1)	Element with the SNMP write view applied to the group.
notify_view	oval-sc:EntityItemStringType (0..1)	Element with the SNMP notify view applied to the group.

< snmphost_item >

Stores information about the SNMP host configuration in IOS. That information includes the host, the community or user strings, the SNMP version, the snmp security (if the SNMP version is SNMPv3) and the SNMP traps.

Extends: oval-sc:ItemType

Child Elements

Table 456: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
host	oval-sc:EntityItemStringType (0..1)	Element with the SNMP host address or hostname.
community_or_user	oval-sc:EntityItemStringType (0..1)	Element with the community string or SNMPv3 user configured for the host.
version	ios-sc:EntityItemSNMPVersionStringType (0..1)	Element with the SNMP version.
snmpv3_sec_level	ios-sc:EntityItemSNMPSecLevelStringType (0..1)	Element with the SNMPv3 security configure for the host.
traps	oval-sc:EntityItemStringType (0..1)	Element with the SNMP traps configured.

< snmpuser_item >

Stores information about an SNMP user configuration in IOS. That information includes the user name, the SNMP group he belongs to, the SNMP version, the IPv4 or IPv6 ACL it is applied to, the Security Level and the Authentication type that apply to the user (for SNMPv3).

Extends: oval-sc:ItemType

Child Elements

Table 457: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	Element with the SNMP user name.
group	oval-sc:EntityItemStringType (0..1)	Element with the SNMP group the user belongs to.
version	ios-sc:EntityItemSNMPVersionStringType (0..1)	Element with the SNMP version of the user.
ipv4_acl	oval-sc:EntityItemStringType (0..1)	Element with the IPv4 ACL name applied to the user.
ipv6_acl	oval-sc:EntityItemStringType (0..1)	Element with the IPv6 ACL name applied to the user.
priv	ios-sc:EntityItemSNMPPrivStringType (0..1)	Element with the SNMP encryption type for the user (for SNMPv3).
auth	ios-sc:EntityItemSNMPAuthStringType (0..1)	Element with the SNMP authentication type for the user (for SNMPv3).

< snmpview_item >

Stores information about an SNMP view configuration in IOS. That information includes the view name, the mib_family that the view uses and the included or excluded option of the mib family in the view.

Extends: oval-sc:ItemType

Child Elements

Table 458: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	Element with the SNMP view name.
mib_family	oval-sc:EntityItemStringType (0..1)	Element with the SNMP MIB family of the view.
include	oval-sc:EntityItemBoolType (0..1)	Element that is true if the included option is used in the view.

< tclsh_item >

The tclsh item holds information about the availability of tcl on the IOS operating system. It extends the standard ItemType as defined in the oval-system-characteristics schema and one should refer to the ItemType description for more information.

Extends: oval-sc:ItemType

Child Elements

Table 459: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
available	oval-sc:EntityItemBoolType (0..1)	This boolean entity describes whether TCLSH is available on the system. A value of true means that TCLSH is available. Per Cisco documentation, the accepted way to see if the device supports tcl functionality is to enter the tcl shell. If the attempt results in a tcl prompt then the device supports tclsh and has it enabled.

< version_item >

The version_item holds information about the version of the IOS operating system. It extends the standard ItemType as defined in the oval-system-characteristics schema and one should refer to the ItemType description for more information.

Extends: oval-sc:ItemType

Child Elements

Table 460: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
major_release (Deprecated)	oval-sc:EntityItemStringType (0..1)	The major_release is a combination of train and rebuild information and is used by Cisco advisories to identify major releases.
train_number (Deprecated)	oval-sc:EntityItemStringType (0..1)	The train number is the dotted version that starts a version string. For example the version string 12.2(3)T has a train number of 12.2.
major_version	oval-sc:EntityItemIntType (0..1)	The major_version entity specifies the major version piece of the version string. The value is an integer and in the example 12.4(9)T0a the major version is '12'.
minor_version	oval-sc:EntityItemIntType (0..1)	The minor_version entity specifies the minor version piece of the version string. The value is an integer and in the example 12.4(9)T0a the minor version is '4'.
release	oval-sc:EntityItemIntType (0..1)	The release entity specifies the release piece of the version string. The value is an integer and in the example 12.4(9)T0a the release is '9'.
train_identifier	oval-sc:EntityItemStringType (0..1)	The train identifier is the type of Train. For example the version string 12.2(3)T has a train identifier of T. Please see the EntityItemTrainIdentifierType for more information about the different train identifiers.The train_identifier entity specifies the type of train represented in the version string. The value is a string and in the example 12.4(9)T0a the train identifier is 'T'. The following explaination from Wikipedia should help explain the different train identifiers. Cisco IOS releases are split into several "trains", each containing a different set of features. Trains more or less map onto distinct markets or groups of customers that Cisco is targeting. The 'mainline' train is designed to be the most stable release the company can offer, and its feature set never expands during its lifetime. Updates are released only to address bugs in the product. The previous technology train becomes the source for the current mainline train–for example, the 12.1T train becomes the basis for the 12.2 mainline. Therefore, to determine the features available in a particular mainline release, look at the previous T train release. The 'T' (Technology) train, gets new features and bug fixes throughout its life, and is therefore less stable than the mainline. (In releases prior to Cisco IOS Release 12.0, the P train served as the Technology train.) The 'S' (Service Provider) train, runs only on the company's core router products and is heavily customized for Service Provider customers. The 'E' (Enterprise) train, is customized for implementation in enterprise environments. The 'B' (broadband) train, support internet based broadband features. The 'XA', 'Xb' … (special functionality) train, needs to be documented. There are other trains from time to time, designed for specific needs – for example, the 12.0AA train contained new code required for Cisco's AS5800 product.
rebuild	oval-sc:EntityItemIntType (0..1)	The rebuild entity specifies the rebuild piece of the version string The value is an integer and in the example 12.4(9)T0a the rebuild is '0'. Often a rebuild is compiled to fix a single specific problem or vulnerability for a given IOS version. For example, 12.1(8)E14 is a Rebuild, the 14 denoting the 14th rebuild of 12.1(8)E. Rebuilds are produced to either quickly repair a defect, or to satisfy customers who do not want to upgrade to a later major revision because they may be running critical infrastructure on their devices, and hence prefer to minimise change and risk.
subrebuild	oval-sc:EntityItemStringType (0..1)	The subrebuild entity specifies the subrebuild piece of the version string. The value is a string and in the example 12.4(9)T0a the subrebuild is 'a'.
mainline_rebuild	oval-sc:EntityItemStringType (0..1)	The mainline_rebuild entity specifies the mainline rebuild piece of the version string. The mainline rebuild is for a regular rebuild release against the mainline operating system release (e.g. the branch of development that would typically be called "the trunk" that isn't associated with a train). Since there is no train identifier to stick the rebuild release after, they stick a alphabetic character inside the parens holding the maintenance release number. For example, 12.4(5b) is the second rebuild of the 12.4(5) maintenance release.

== EntityItemAccessListInterfaceDirectionType ==

The EntityItemAccessListInterfaceDirectionType complex type restricts a string value to a specific set of values: IN, OUT. These values describe the inbound or outbound ACL direction on an interface in a Cisco IOS configuration. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 461: Enumeration Values

Value	Description
IN	(No Description)
OUT	(No Description)
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemAccessListIPVersionType ==

The EntityItemRoutingProtocolType complex type restricts a string value to a specific set of values: IPV4, IPV6. These values describe if an ACL is for IPv4 or IPv6 in a Cisco IOS configuration. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 462: Enumeration Values

Value	Description
IPV4	(No Description)
IPV6	(No Description)
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemAccessListUseType ==

The EntityItemAccessListUseType complex type restricts a string value to a specific set of values: INTERFACE, CRYPTO_MAP_MATCH, CLASS_MAP_MATCH, ROUTE_MAP_MATCH, IGMP_FILTER, VTY. These values describe the ACL use in a Cisco IOS configuration. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 463: Enumeration Values

Value	Description
INTERFACE	(No Description)
CRYPTO_MAP_MATCH	(No Description)
CLASS_MAP_MATCH	(No Description)
ROUTE_MAP_MATCH	(No Description)
IGMP_FILTER	(No Description)
VTY	(No Description)
NONE (Deprecated)	**Deprecated As Of Version:** 5.11.2:1.0 **Reason:** The EntityStateSimpleBaseType check_existence attribute serves the same purpose as this enumeration value. **Comment:** This AccessListUseType enumeration value has been deprecated and may be removed in a future version of the language.
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemRoutingAuthTypeStringType ==

The EntityItemRoutingAuthTypeStringType complex type restricts a string value to a specific set of values: CLEARTEXT, MESSAGE_DIGEST. These values describe the routing protocol authentication types used in a Cisco IOS configuration. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 464: Enumeration Values

Value	Description
CLEARTEXT	(No Description)
MESSAGE_DIGEST	(No Description)
NULL (Deprecated)	**Deprecated As Of Version:** 5.11.2:1.0 **Reason:** The NULL authentication area type is never declared in an interface ip ospf command context. **Comment:** This RoutingAuthTypeStringType enumeration value has been deprecated and may be removed in a future version of the language.
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemRoutingProtocolType ==

The EntityItemRoutingProtocolType complex type restricts a string value to a specific set of values: EIGRP, OSPF, BGP, RIP, RIPV2, ISIS. These values describe the routing protocol used in a Cisco IOS configuration. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 465: Enumeration Values

Value	Description
EIGRP	(No Description)
OSPF	(No Description)
BGP	(No Description)
RIP	(No Description)
RIPV2	(No Description)
ISIS	(No Description)
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemSNMPVersionStringType ==

The EntityItemSNMPVersionStringType complex type restricts a string value to a specific set of values: 1, 2c, 3. These values describe the SNMP version in a Cisco IOS configuration. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 466: Enumeration Values

Value	Description
1	(No Description)
2C	(No Description)
3	(No Description)
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemSNMPSecLevelStringType ==

The EntityItemSNMPVersionStringType complex type restricts a string value to a specific set of values: PRIV, AUTH, NO_AUTH. These values describe the SNMP security level (encryption, Authentication, None) in a Cisco IOS SNMPv3 related configurations. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 467: Enumeration Values

Value	Description
PRIV	(No Description)
AUTH	(No Description)
NO_AUTH	(No Description)
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemSNMPModeStringType ==

The EntityItemSNMPModeStringType complex type restricts a string value to a specific set of values: RO, RW. These values describe the SNMP mode (read-only, read-write) in a Cisco IOS SNMPv3 related configurations. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 468: Enumeration Values

Value	Description
RO	(No Description)
RW	(No Description)
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemSNMPAuthStringType ==

The EntityItemSNMPAuthStringType complex type restricts a string value to a specific set of values: MD5, SHA. These values describe the authentication algorithm in a Cisco IOS SNMPv3 related configurations. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 469: Enumeration Values

Value	Description
MD5	(No Description)
SHA	(No Description)
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemSNMPPrivStringType ==

The EntityItemSNMPPrivStringType complex type restricts a string value to a specific set of values: DES, 3DES, AES. These values describe the encryption algorithm in a Cisco IOS SNMPv3 related configurations. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 470: Enumeration Values

Value	Description
DES	(No Description)
3DES	(No Description)
AES	(No Description)
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemSwitchportModeType ==

The EntityItemRoutingProtocolType complex type restricts a string value to a specific set of values: DYNAMIC, TRUNK, ACCESS. These values describe the interface switchport mode types in IOS. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 471: Enumeration Values

Value	Description
DYNAMIC	(No Description)
TRUNK	(No Description)
ACCESS	(No Description)
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemTrunkEncapType ==

The EntityItemTrunkEncapType complex type restricts a string value to a specific set of values: DOT1Q, ISL, NEGOTIATE. These values describe the interface trunk encapsulation types on an interfaces in IOS. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 472: Enumeration Values

Value	Description
DOT1Q	(No Description)
ISL	(No Description)
NEGOTIATE	(No Description)
	The empty string value is permitted here to allow for detailed error reporting.

Open Vulnerability and Assessment Language: IOS-XE Definition

- Schema: IOS-XE Definition
- Version: 5.11.1:1.2
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the IOS-XE specific tests found in Open Vulnerability and Assessment Language (OVAL). Each test is an extension of the standard test element defined in the Core Definition Schema. Through extension, each test inherits a set of elements and attributes that are shared amongst all OVAL tests. Each test is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core Definition Schema is not outlined here.

The OVAL Schema is maintained by the 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.cisecurity.org.

Thanks to Omar Santos and Panos Kampanakis of Cisco for providing this test.

Test Listing

- *< global_test >*
- *< line_test >*
- *< version_test >*
- *< interface_test >*
- *< section_test >*
- *< router_test >*
- *< bgpneighbor_test >*
- *< routingprotocolauthintf_test >*
- *< acl_test >*
- *< snmphost_test >*
- *< snmpcommunity_test >*
- *< snmpuser_test >*
- *< snmpgroup_test >*

- *< snmpview_test >*

< global_test >

The global test is used to check for the existence of a particular line in the IOS-XE config file under the global context. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a global_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 473: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< global_object >

The global_object element is used by a global test to define the object to be evaluated. For the most part this object checks for existence and is used without a state comparision. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 474: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
global_command	oval-def:EntityObjectStringType (1..1)	The global_command entity identifies a specific line in the IOS-XE config file under the global context.
oval-def:filter	n/a (0..unbounded)	

< global_state >

The global_state element defines the different information that can be found in the IOS-XE config file under the global context. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 475: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
global_command	oval-def:EntityStateStringType (0..1)	The global_command entity identifies a specific line in the IOS-XE config file under the global context.

< line_test >

The line test is used to check the properties of specific output lines from a SHOW command, such as show running-config. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a line_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 476: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< line_object >

The line_object element is used by a line test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A line object consists of a show_subcommand entity that is the name of a SHOW sub-command to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 477: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
show_subcommand	oval-def:EntityObjectStringType (1..1)	The name of a SHOW sub-command.
oval-def:filter	n/a (0..unbounded)	

< line_state >

The line_state element defines the different information that can be used to evaluate the result of a specific SHOW sub-command. This includes the name of ths sub-command and the corresponding config line. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 478: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
show_subcommand	oval-def:EntityStateStringType (0..1)	The name of the SHOW sub-command.
config_line	oval-def:EntityStateStringType (0..1)	The value returned from by the specified SHOW sub-command.

< version_test >

The version_test is used to check the version of the IOS-XE operating system. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a version_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 479: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< version_object >

The version_object element is used by a version_test to define the different version information associated with an IOS-XE system. There is actually only one object relating to version and this is the system as a whole. Therefore, there are no child entities defined. Any OVAL Test written to check version will reference the same version_object which is basically an empty object element.

Extends: oval-def:ObjectType

< version_state >

The version_state element defines the version information held within a Cisco IOS-XE Train. A Cisco IOS-XE train is a vehicle for delivering releases that evolve from a common code base.

Extends: oval-def:StateType

Child Elements

Table 480: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
platform (Deprecated)	oval-def:EntityStateStringType (0..1)	The platform that is running the IOS-XE software. For example if could be asr1000.
rp (Deprecated)	oval-def:EntityStateIntType (0..1)	The routing processor running the IOS-XE software.
pkg (Deprecated)	oval-def:EntityStateStringType (0..1)	The consolidated IOS-XE packages in the image. For example it could be advservicesk9.
version_string	oval-def:EntityStateStringType (0..1)	The entire IOS-XE version string, for example, '03.13.02.S'.
major_release	oval-def:EntityStateIntType (0..1)	The major version piece of the version string. The value is an integer, and in the example 03.13.02.S the major_release is '3'
release	oval-def:EntityStateIntType (0..1)	The minor release piece of the version string. The value is an integer, and in the example 03.13.02.S the release is '13'
rebuild	oval-def:EntityStateIntType (0..1)	The rebuild piece of the version string. The value is an integer, and in the example 03.13.02.S the rebuild is '2'
train	oval-def:EntityStateStringType (0..1)	The train piece of the version string. The value is a string, and in the example 03.13.02.S the train is 'S'
ios_release (Deprecated)	oval-def:EntityStateStringType (0..1)	The IOS release the IOS-XE was derived from. The value is a string and in the example ASR1000rp1-ipbasek9.03.04.02.122-33.SR.bin the ios_release version is '122-33'
ios_train (Deprecated)	oval-def:EntityStateStringType (0..1)	The IOS release the IOS-XE was derived from. The value is an integer and in the example ASR1000rp1-ipbasek9.03.04.02.122-33.SR.bin the ios_release version is 'SR'

< interface_test >

The interface test is used to check for the existence of a particular interface on the Cisco IOS-XE device. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a interface_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 481: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< interface_object >

The interface_object element is used by an interface_test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An interface_object consists of a name entity that is the name of the IOS-XE interface to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 482: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityObjectStringType (1..1)	The interface name.
oval-def:filter	n/a (0..unbounded)	

< interface_state >

The interface_state element defines the different information that can be used to evaluate the result of a specific IOS-XE interface. This includes the name, status, and address information about the interface. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 483: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	The interface name.
ip_directed_broadcast	oval-def:EntityStateBoolType (0..1)	Directed broadcast command enabled on the interface. The default is false.
proxy_arp	oval-def:EntityStateBoolType (0..1)	Proxy arp enabled on the interface. The default is true.
shutdown	oval-def:EntityStateBoolType (0..1)	Interface is shut down.
hardware_addr	oval-def:EntityStateStringType (0..1)	The interface hardware (MAC) address.
ipv4_address	oval-def:EntityStateIPAddressStringType (0..1)	The interface IPv4 address and mask. This element should only allow 'ipv4_address' of the oval:SimpleDatatypeEnumeration.
ipv6_address	oval-def:EntityStateIPAddressStringType (0..1)	The interface IPv6 address and mask. This element should only allow 'ipv6_address' of the oval:SimpleDatatypeEnumeration.
ipv4_access_list	oval-def:EntityStateStringType (0..1)	The ingress or egress IPv4 ACL name applied on the interface.
ipv6_access_list	oval-def:EntityStateStringType (0..1)	The ingress or egress IPv6 ACL name applied on the interface.
crypto_map	oval-def:EntityStateStringType (0..1)	The crypto map name applied to the interface.
ipv4_urpf_command	oval-def:EntityStateStringType (0..1)	The IPv4 uRPF command under the interface.
ipv6_urpf_command	oval-def:EntityStateStringType (0..1)	The IPv6 uRPF command under the interface.
urpf_command (Deprecated)	oval-def:EntityStateStringType (0..1)	The uRPF command under the interface.
switchport_trunk_encapsulation	iosxe-def:EntityStateTrunkEncapType (0..1)	The switchport trunk encapsulation option configured on the interface (if applicable).
switchport_mode	iosxe-def:EntityStateSwitchportModeType (0..1)	The switchport mode option configured on the interface (if applicable).
switchport_native_vlan	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	The trunk native vlan configured on the interface (if applicable).
switchport_access_vlan	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	The access vlan configured on the interface (if applicable).
switchport_trunked_vlans	oval-def:EntityStateStringType (0..1)	The vlans that are trunked configured on the interface (if applicable).
switchport_pruned_vlans	oval-def:EntityStateStringType (0..1)	The vlans that are pruned from the trunk (if applicable).
switchport_port_security	oval-def:EntityStateStringType (0..1)	The switchport port-security commands configured on the interface (if applicable).

< section_test >

The section test is used to check the properties of specific output lines from a configuration section.

Extends: oval-def:TestType

Child Elements

Table 484: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< section_object >

The section_object element is used by a section test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A section object consists of a section_command entity that is the name of a section command to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 485: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
section_command	oval-def:EntityObjectStringType (1..1)	The name of a section command.
oval-def:filter	n/a (0..unbounded)	

< section_state >

The section_state element defines the different information that can be used to evaluate the result of a specific section command. This includes the name of ths section_command and the corresponding config lines. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 486: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
section_command	oval-def:EntityStateStringType (0..1)	The name of the section command.
section_config_lines	oval-def:EntityStateStringType (0..1)	The value returned with all config lines of the section.
config_line	oval-def:EntityStateStringType (0..1)	The value returned with one config line of the section at a time.

< router_test >

The router test is used to check the properties of specific output lines from a router configurated instance in IOS-XE.

Extends: oval-def:TestType

Child Elements

Table 487: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< router_object >

The router_object element is used by a router test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A router object consists of a router protocol and router identifier entity.

Extends: oval-def:ObjectType

Child Elements

Table 488: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
protocol	iosxe-def:EntityObjectRoutingProtocolType (1..1)	The routing protocol of the router instance.
id	oval-def:EntityObjectIntType (1..1)	The IOS-XE router id.
oval-def:filter	n/a (0..unbounded)	

< router_state >

The router_state element defines the different information that can be used to evaluate the result of a specific router command. This includes the protocol of the router instance, the id, the networks, bgp neighbor, ospf authentication area commands and the corresponding config lines. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 489: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
protocol	iosxe-def:EntityStateRoutingProtocolType (1..1)	The routing protocol of the router instance. If there are more than one router configurations, for example ospf instances, different objects should be created for each.
id	oval-def:EntityStateIntType (0..1)	The IOS-XE router id
network	oval-def:EntityStateStringType (0..1)	The subnet in the network command of the router instance. The area can be included in the string for OSPF.
bgp_neighbor	oval-def:EntityStateStringType (0..1)	The BGP neighbors, if applicable.
ospf_authentication_area	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	The OSPF area that is authenticated, if applicable.
router_config_lines	oval-def:EntityStateStringType (0..1)	The value returned with all config lines of the router instance.

< bgpneighbor_test >

The bgpneighbor test is used to check the bgp neighbpr properties of bgp instances instances in IOS.

Extends: oval-def:TestType

Child Elements

Table 490: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< bgpneighbor_object >

The bgpneighbor_object element is used by a bgpneighbor test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A bgpneighbor object consists of a neighbor entity.

Extends: oval-def:ObjectType

Child Elements

Table 491: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
neighbor	oval-def:EntityObjectStringType (1..1)	The bgp neighbor.
oval-def:filter	n/a (0..unbounded)	

< bgpneighbor_state >

The bgpneighbor_state element defines the different information that can be used to evaluate the result of a bgp neighbor configuration. This includes the neighbor and the password option, if configured. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 492: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
neighbor	oval-def:EntityStateStringType (0..1)	The bgp neighbor.
password	oval-def:EntityStateStringType (0..1)	The bgp authentication password, if configured. If Encryption type is configured it should be included in the password string. For example '0 cisco123'.

< routingprotocolauthintf_test >

The routing protocol authentication interface test is used to check the properties of routing protocol authentication configured under interfaces in IOS.

Extends: oval-def:TestType

Child Elements

Table 493: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< routingprotocolauthintf_object >

The routingprotocolauthintf_object element is used by a routingprotocolauthintf test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A routingprotocolauthintf object consists of an interface and the routing protocol that is authenticated entity.

Extends: oval-def:ObjectType

Child Elements

Table 494: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
interface	oval-def:EntityObjectStringType (1..1)	The interface name.
protocol	iosxe-def:EntityObjectRoutingProtocolType (1..1)	The routing protocol.
oval-def:filter	n/a (0..unbounded)	

< routingprotocolauthintf_state >

The routingprotocolauthintf_state element defines the different information that can be used to evaluate the result of a specific routing protocol interface authentication configurations. This includes the interface, the protocol, the id, the authentication type, the ospf area, the key chain command and the corresponding config lines. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 495: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
interface	oval-def:EntityStateStringType (0..1)	The interface name.
protocol	iosxe-def:EntityStateRoutingProtocolType (0..1)	The routing protocol.
id	oval-def:EntityStateIntType (0..1)	The routing protocol id, if applicable.
auth_type	iosxe-def:EntityStateRoutingAuthTypeStringType (0..1)	The routing protocol authentication type.
ospf_area	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	The OSPF area that is authenticated, if applicable.
key_chain	oval-def:EntityStateStringType (0..1)	The name of the key chain, if applicable.

< acl_test >

The acl test is used to check the properties of specific output lines from an ACL configuration.

Extends: oval-def:TestType

Child Elements

Table 496: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< acl_object >

The acl_object element is used by an acl test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An acl object consists of a an acl name and an IP version entity that is the name and the IP protocol version of the access-list to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 497: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityObjectStringType (1..1)	The name of the ACL.
ip_version	iosxe-def:EntityObjectAccessListIPVersionType (1..1)	The IP version of the ACL.
oval-def:filter	n/a (0..unbounded)	

< acl_state >

The acl_state element defines the different information that can be used to evaluate the result of a specific ACL configuration. This includes the name of ths ACL and the corresponding config lines. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 498: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	The name of the ACL.
ip_version	iosxe-def:EntityStateAccessListIPVersionType (0..1)	The IP version of the ACL.
use	iosxe-def:EntityStateAccessListUseType (0..1)	The feature where the ACL is used.
used_in	oval-def:EntityStateStringType (0..1)	The name of where the ACL is used. For example if use is 'INTERFACE', use_in will be the name of the interface.
interface_direction	iosxe-def:EntityStateAccessListInterfaceDirectionType (0..1)	The direction the ACL is applied on an interface.
acl_config_lines	oval-def:EntityStateStringType (0..1)	The value returned with all config lines of the ACL.
config_line	oval-def:EntityStateStringType (0..1)	The value returned with one ACL config line at a time.

< snmphost_test >

The snmphost test is used to check the properties of specific output lines from an SNMP configuration.

Extends: oval-def:TestType

Child Elements

Table 499: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< snmphost_object >

The snmphost_object element is used by an snmphost test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A snmphost object consists of a host entity that is the host of the 'snmp host' IOS-XE command to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 500: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
host	oval-def:EntityObjectStringType (1..1)	The SNMP host address or hostname.
oval-def:filter	n/a (0..unbounded)	

< snmphost_state >

The snmphost_state element defines the different information that can be used to evaluate the result of a specific 'snmp host' IOS-XE command. This includes the host and the corresponding options. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 501: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
host	oval-def:EntityStateStringType (0..1)	The SNMP host address or hostname.
community_or_user	oval-def:EntityStateStringType (0..1)	The community string or SNMPv3 user configured for the host.
version	iosxe-def:EntityStateSNMPVersionStringType (0..1)	The SNMP version.
snmpv3_sec_level	iosxe-def:EntityStateSNMPSecLevelStringType (0..1)	The SNMPv3 security configured for the host.
traps	oval-def:EntityStateStringType (0..1)	The SNMP traps configured.

< snmpcommunity_test >

The snmpcommunity test is used to check the properties of specific output lines from an SNMP configuration.

Extends: oval-def:TestType

Child Elements

Table 502: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< snmpcommunity_object >

The snmpcommunity_object element is used by an snmpcommunity test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An snmpcommunity object consists of a community name entity to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 503: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityObjectStringType (1..1)	The SNMP community name.
oval-def:filter	n/a (0..unbounded)	

< snmpcommunity_state >

The snmpcommunity_state element defines the different information that can be used to evaluate the result of a specific ‘snmp community’ IOS-XE command. This includes the community name and the corresponding options. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 504: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	The SNMP community name.
view	oval-def:EntityStateStringType (0..1)	The view that restricts the OIDs of this community.
mode	iosxe-def:EntityStateSNMPModeStringType (0..1)	The read-write privileges of the community.
ipv4_acl	oval-def:EntityStateStringType (0..1)	The IPv4 ACL name applied to the community.
ipv6_acl	oval-def:EntityStateStringType (0..1)	The IPv6 ACL name applied to the community.

< snmpuser_test >

The snmpuser test is used to check the properties of specific output lines from an SNMP user configuration.

Extends: oval-def:TestType

Child Elements

Table 505: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< snmpuser_object >

The snmpuser_object element is used by an snmpuser test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A snmpuser object consists of a name entity that is the name of the SNMP user to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 506: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityObjectStringType (1..1)	The SNMP user name.
oval-def:filter	n/a (0..unbounded)	

< snmpuser_state >

The snmpuser_state element defines the different information that can be used to evaluate the result of a specific 'show snmp user' IOS-XE command. This includes the user name and the corresponding options. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 507: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	The SNMP user name.
group	oval-def:EntityStateStringType (0..1)	The SNMP group the user belongs to.
version	iosxe-def:EntityStateSNMPVersionStringType (0..1)	The SNMP version of the user.
ipv4_acl	oval-def:EntityStateStringType (0..1)	The IPv4 ACL name applied to the user.
ipv6_acl	oval-def:EntityStateStringType (0..1)	The IPv6 ACL name applied to the user.
priv	iosxe-def:EntityStateSNMPPrivStringType (0..1)	The SNMP encryption type for the user (for SNMPv3).
auth	iosxe-def:EntityStateSNMPAuthStringType (0..1)	The SNMP authentication type for the user (for SNMPv3).

< snmpgroup_test >

The snmpgroup test is used to check the properties of specific output lines from an SNMP group configuration.

Extends: oval-def:TestType

Child Elements

Table 508: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< snmpgroup_object >

The snmpgroup_object element is used by an snmpgroup test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A snmpgroup object consists of a name entity that is the name of the SNMP group to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 509: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityObjectStringType (1..1)	The SNMP group name.
oval-def:filter	n/a (0..unbounded)	

< snmpgroup_state >

The snmpgroup_state element defines the different information that can be used to evaluate the result of a specific 'snmp-server group' IOS-XE command. This includes the user name and the corresponding options. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 510: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	The SNMP group name.
version	iosxe-def:EntityStateSNMPVersionStringType (0..1)	The SNMP version of the group.
snmpv3_sec_level	iosxe-def:EntityStateSNMPSecLevelStringType (0..1)	The SNMPv3 security configured for the group.
ipv4_acl	oval-def:EntityStateStringType (0..1)	The IPv4 ACL name applied to the group.
ipv6_acl	oval-def:EntityStateStringType (0..1)	The IPv6 ACL name applied to the group.
read_view	oval-def:EntityStateStringType (0..1)	The SNMP read view applied to the group.
write_view	oval-def:EntityStateStringType (0..1)	The SNMP write view applied to the group.
notify_view	oval-def:EntityStateStringType (0..1)	The SNMP notify view applied to the group.

< snmpview_test >

The snmpview test is used to check the properties of specific output lines from an SNMP view configuration.

Extends: oval-def:TestType

Child Elements

Table 511: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< snmpview_object >

The snmpview_object element is used by an snmpview test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A snmpview object consists of a name entity that is the name of the SNMP view to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 512: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityObjectStringType (1..1)	The SNMP view name.
oval-def:filter	n/a (0..unbounded)	

< snmpview_state >

The snmpview_state element defines the different information that can be used to evaluate the result of a specific ‘snmp-server view’ IOS-XE command. This includes the view name and the corresponding options. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 513: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	The SNMP view name.
mib_family	oval-def:EntityStateStringType (0..1)	The SNMP MIB family of the view.
include	oval-def:EntityStateBoolType (0..1)	It is true if the included option is used in the view.

== EntityObjectAccessListIPVersionType ==

The EntityObjectAccessListIPVersionType complex type restricts a string value to a specific set of values: IPV4, IPV6. These values describe if an ACL is for IPv4 or IPv6 in a Cisco IOS-XE configuration. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityObjectStringType

Table 514: Enumeration Values

Value	Description
IPV4	(No Description)
IPV6	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityObjectRoutingProtocolType ==

The EntityObjectRoutingProtocolType complex type restricts a string value to a specific set of values: EIGRP, OSPF, BGP, RIP, RIPV2, ISIS. These values describe the routing protocol used in a Cisco IOS-XE configuration. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityObjectStringType

Table 515: Enumeration Values

Value	Description
EIGRP	(No Description)
OSPF	(No Description)
BGP	(No Description)
RIP	(No Description)
RIPV2	(No Description)
ISIS	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateTrunkEncapType ==

The EntityStateTrunkEncapType complex type restricts a string value to a specific set of values: DOT1Q, ISL, NEGOTIATE. These values describe the interface trunk encapsulation types on an interfaces in IOS. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 516: Enumeration Values

Value	Description
DOT1Q	(No Description)
ISL	(No Description)
NEGOTIATE	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateSwitchportModeType ==

The EntityStateSwitchportModeType complex type restricts a string value to a specific set of values: DYNAMIC, TRUNK, ACCESS. These values describe the interface switchport mode types in IOS. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 517: Enumeration Values

Value	Description
DYNAMIC	(No Description)
TRUNK	(No Description)
ACCESS	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateRoutingProtocolType ==

The EntityStateRoutingProtocolType complex type restricts a string value to a specific set of values: EIGRP, OSPF, BGP, RIP, RIPV2, ISIS. These values describe the routing protocol used in a Cisco IOS-XE configuration. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 518: Enumeration Values

Value	Description
EIGRP	(No Description)
OSPF	(No Description)
BGP	(No Description)
RIP	(No Description)
RIPV2	(No Description)
ISIS	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateRoutingAuthTypeStringType ==

The EntityStateRoutingAuthTypeStringType complex type restricts a string value to a specific set of values: CLEARTEXT, MESSAGE_DIGEST. These values describe the routing protocol authentication types used in a Cisco IOS-XE configuration. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 519: Enumeration Values

Value	Description
CLEARTEXT	(No Description)
MESSAGE_DIGEST	(No Description)
NULL (Deprecated)	**Deprecated As Of Version:** 5.11.2:1.0 **Reason:** The NULL authentication area type is never declared in an interface ip ospf command context. **Comment:** This RoutingAuthTypeStringType enumeration value has been deprecated and may be removed in a future version of the language.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateSNMPVersionStringType ==

The EntityStateSNMPVersionStringType complex type restricts a string value to a specific set of values: 1, 2c, 3. These values describe the SNMP version in a Cisco IOS-XE configuration. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 520: Enumeration Values

Value	Description
1	(No Description)
2C	(No Description)
3	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateSNMPSecLevelStringType ==

The EntityStateSNMPSecLevelStringType complex type restricts a string value to a specific set of values: PRIV, AUTH, NO_AUTH. These values describe the SNMP security level (encryption, Authentication, None) in a Cisco IOS-XE SNMPv3 related configurations. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 521: Enumeration Values

Value	Description
PRIV	(No Description)
AUTH	(No Description)
NO_AUTH	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateSNMPModeStringType ==

The EntityStateSNMPModeStringType complex type restricts a string value to a specific set of values: RO, RW. These values describe the SNMP mode (read-only, read-write) in a Cisco IOS-XE SNMPv3 related configurations. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 522: Enumeration Values

Value	Description
RO	(No Description)
RW	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateSNMPAuthStringType ==

The EntityStateSNMPAuthStringType complex type restricts a string value to a specific set of values: MD5, SHA. These values describe the authentication algorithm in a Cisco IOS-XE SNMPv3 related configurations. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 523: Enumeration Values

Value	Description
MD5	(No Description)
SHA	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateSNMPPrivStringType ==

The EntityStateSNMPPrivStringType complex type restricts a string value to a specific set of values: DES, 3DES, AES. These values describe the encryption algorithm in a Cisco IOS-XE SNMPv3 related configurations. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 524: Enumeration Values

Value	Description
DES	(No Description)
3DES	(No Description)
AES	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateAccessListIPVersionType ==

The EntityStateAccessListIPVersionType complex type restricts a string value to a specific set of values: IPV4, IPV6. These values describe if an ACL is for IPv4 or IPv6 in a Cisco IOS-XE configuration. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 525: Enumeration Values

Value	Description
IPV4	(No Description)
IPV6	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateAccessListUseType ==

The EntityStateAccessListUseType complex type restricts a string value to a specific set of values: INTERFACE, CRYPTO_MAP_MATCH, CLASS_MAP_MATCH, ROUTE_MAP_MATCH, IGMP_FILTER, VTY. These values describe the ACL use in a Cisco IOS-XE configuration. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 526: Enumeration Values

Value	Description
INTERFACE	(No Description)
CRYPTO_MAP_MATCH	(No Description)
CLASS_MAP_MATCH	(No Description)
ROUTE_MAP_MATCH	(No Description)
IGMP_FILTER	(No Description)
VTY	(No Description)
NONE (Deprecated)	**Deprecated As Of Version:** 5.11.2:1.0 **Reason:** The EntityStateSimpleBaseType check_existence attribute serves the same purpose as this enumeration value. **Comment:** This AccessListUseType enumeration value has been deprecated and may be removed in a future version of the language.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateAccessListInterfaceDirectionType ==

The EntityStateAccessListInterfaceDirectionType complex type restricts a string value to a specific set of values: IN, OUT. These values describe the inbound or outbound ACL direction on an interface in a Cisco IOS-XE configuration. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 527: Enumeration Values

Value	Description
IN	(No Description)
OUT	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

Open Vulnerability and Assessment Language: IOS-XE System Characteristics

- Schema: IOS-XE System Characteristics
- Version: 5.11.1:1.2
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the IOS-XE specific system characteristic items found in Open Vulnerability and Assessment Language (OVAL). Each item is an extension of the standard item element defined in the Core System Characteristic Schema. Through extension, each item inherits a set of elements and attributes that are shared amongst all OVAL Items. Each item is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core System Characteristic Schema is not outlined here.

The OVAL Schema is maintained by the 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.cisecurity.org.

Thanks to Omar Santos and Panos Kampanakis of Cisco for providing this test.

Item Listing

- *< global_item >*
- *< line_item >*
- *< version_item >*
- *< section_item >*
- *< interface_item >*
- *< router_item >*
- *< bgpneighbor_item >*
- *< routingprotocolauthintf_item >*
- *< acl_item >*
- *< snmphost_item >*
- *< snmpcommunity_item >*
- *< snmpuser_item >*
- *< snmpgroup_item >*

- *< snmpview_item >*

< global_item >

Sotres information about the existence of a particular line in the IOS-XE config file under the global context

Extends: oval-sc:ItemType

Child Elements

Table 528: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
global_command	oval-sc:EntityItemStringType (0..1)	The global_command entity identifies a specific line in the IOS-XE config file under the global context.

< line_item >

Stores the properties of specific lines in the IOS-XE config file.

Extends: oval-sc:ItemType

Child Elements

Table 529: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
show_subcommand	oval-sc:EntityItemStringType (0..1)	The name of the SHOW sub-command.
config_line	oval-sc:EntityItemStringType (0..1)	The value returned from by the specified SHOW sub-command.

< version_item >

The version_item holds information about the version of the IOS-XE operating system. It extends the standard ItemType as defined in the oval-system-characteristics schema and one should refer to the ItemType description for more information.

Extends: oval-sc:ItemType

Child Elements

Table 530: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
platform (Deprecated)	oval-sc:EntityItemStringType (0..1)	The platform entity specifies the platform that is running the IOS-XE software. For example if could be asr1000.
rp (Deprecated)	oval-sc:EntityItemIntType (0..1)	The rp entity specifies the routing processor running the IOS-XE software.
pkg (Deprecated)	oval-sc:EntityItemStringType (0..1)	The pkg entity specifies the consolidated IOS-XE packages in the image. For example it could be adventservicesk9.
version_string	oval-sc:EntityItemStringType (0..1)	The train entity specifies the entire IOS-XE version string, for example, '03.13.02.S'.
major_release	oval-sc:EntityItemIntType (0..1)	The major_release entity specifies the major version piece of the version string. The value is an integer and in the example 03.13.02.S the major_release is '3'.
release	oval-sc:EntityItemIntType (0..1)	The release entity specifies the release piece of the version string. The value is an integer and in the example 03.13.02.S the release version is '13'.
rebuild	oval-sc:EntityItemIntType (0..1)	The rebuild entity specifies the release piece of the version string. The value is an integer and in the example 03.13.02.S the rebuild is '2'.
train	oval-sc:EntityItemStringType (0..1)	The train entity specifies the train piece of the version string. The value is a string and in the example 03.13.02.S the train is 'S'.
ios_release (Deprecated)	oval-sc:EntityItemStringType (0..1)	The ios_release entity specifies the IOS release the IOS-XE was derived from. The value is an string and in the example ASR1000rp1-ipbasek9.03.04.02.122-33.SR.bin the ios_release version is '122-33'
ios_train (Deprecated)	oval-sc:EntityItemStringType (0..1)	The ios_train entity specifies the IOS release the IOS-XE was derived from. The value is an integer and in the example ASR1000rp1-ipbasek9.03.04.02.122-33.SR.bin the ios_release version is 'SR'

< section_item >

Stores command that are part of a IOS-XE configuration section. For example all configuration lines under an interface. It should not store configurations for configs that already have a separate item. For example BGP has a router item and should not also be stored in a section_item.

Extends: oval-sc:ItemType

Child Elements

Table 531: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
section_command	oval-sc:EntityItemStringType (0..1)	The name of the section command.
section_config_lines	oval-sc:EntityItemStringType (0..1)	Element with all config lines of the section
config_line	oval-sc:EntityItemStringType (0..unbounded)	Element with one config line of the section at a time

---

< interface_item >

The interface_item represents an IOS-XE interface and its configuration options.

Extends: oval-sc:ItemType

Child Elements

Table 532: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	Element with the interface name.
ip_directed_broadcast	oval-sc:EntityItemBoolType (0..1)	Element that is true if the directed broadcast command is enabled on the interface. The default is false.
proxy_arp	oval-sc:EntityItemBoolType (0..1)	Element that is true if the proxy_arp command is enabled on the interface. The default is true.
shutdown	oval-sc:EntityItemBoolType (0..1)	Element that is true if the interface is shut down. The default is false.
hardware_addr	oval-sc:EntityItemStringType (0..1)	Element with the interface hardware (MAC) address.
ipv4_address	oval-sc:EntityItemIPAddressStringType (0..1)	Element with the interface IPv4 address and mask. This element should only allow 'ipv4_address' of the oval:SimpleDatatypeEnumeration.
ipv6_address	oval-sc:EntityItemIPAddressStringType (0..unbounded)	Element with the interface IPv6 address and mask. This element should only allow 'ipv6_address' of the oval:SimpleDatatypeEnumeration.
ipv4_access_list	oval-sc:EntityItemStringType (0..2)	Element with the ingress or egress IPv4 ACL name applied on the interface.
ipv6_access_list	oval-sc:EntityItemStringType (0..2)	Element with the ingress or egress IPv6 ACL name applied on the interface.
crypto_map	oval-sc:EntityItemStringType (0..1)	Element with the crypto map name applied to the interface.
ipv4_urpf_command	oval-sc:EntityItemStringType (0..1)	Element with the uRPF command for IPv4 under the interface.
ipv6_urpf_command	oval-sc:EntityItemStringType (0..1)	Element with the uRPF command for IPv6 under the interface.
urpf_command (Deprecated)	oval-sc:EntityItemStringType (0..1)	Element with the uRPF command under the interface.
switchport_trunk_encapsulation	iosxe-sc:EntityItemTrunkEncapType (0..1)	Element with the switchport trunk encapsulation option configured on the interface (if applicable).
switchport_mode	iosxe-sc:EntityItemSwitchportModeType (0..1)	Element with the switchport mode option configured on the interface (if applicable).
switchport_native_vlan	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	Element with the trunk native vlan configured on the interface (if applicable).
switchport_access_vlan	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	Element with the access vlan configured on the interface (if applicable).
switchport_trunked_vlans	oval-sc:EntityItemStringType (0..1)	Element with the vlans that are trunked configured on the interface (if applicable).
switchport_pruned_vlans	oval-sc:EntityItemStringType (0..1)	Element with the vlans that are pruned from the trunk (if applicable).
switchport_port_security	oval-sc:EntityItemStringType (0..1)	Element with the switchport port-security commands configured on the interface (if applicable).

< router_item >

Stores commands that are part of a IOS-XE ‘router’ command configuration. For example ‘router bgp 123’.

Extends: oval-sc:ItemType

Child Elements

Table 533: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
protocol	iosxe-sc:EntityItemRoutingProtocolType (0..1)	Element with the routing protocol.
id	oval-sc:EntityItemIntType (0..1)	Element with the IOS-XE router id.
network	oval-sc:EntityItemStringType (0..unbounded)	Element with the subnet in the network command of the router instance. The area can be included in the string for OSPF.
bgp_neighbor	oval-sc:EntityItemStringType (0..unbounded)	Element with the BGP neighbors, if applicable.
ospf_authentication_area	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..unbounded)	Element with the OSPF area that is authenticated, if applicable.
router_config_lines	oval-sc:EntityItemStringType (0..1)	Element with all config lines of the router.

< bgpneighbor_item >

Stores information about bgp neighbors configured in bgp instances.

Extends: oval-sc:ItemType

Child Elements

Table 534: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
neighbor	oval-sc:EntityItemStringType (0..1)	Element with the bgp neighbor.
password	oval-sc:EntityItemStringType (0..1)	Element with the bgp authentication password, if configured. If Encryption type is configured it should be included in the password string. For example ‘0 cisco123’.

< routingprotocolauthintf_item >

Stores information for routing protocol authentication configured under specific interfaces.

Extends: oval-sc:ItemType

Child Elements

Table 535: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
interface	oval-sc:EntityItemStringType (0..1)	Element with the interface.
protocol	iosxe-sc:EntityItemRoutingProtocolType (0..1)	Element with the routing protocol.
id	oval-sc:EntityItemIntType (0..1)	Element with the routing protocol id.
auth_type	iosxe-sc:EntityItemRoutingAuthTypeStringType (0..1)	Element with the routing protocol authentication type.
ospf_area	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	Element with the OSPF area that is authenticated, if applicable.
key_chain	oval-sc:EntityItemStringType (0..1)	Element with the name of the key chain, if applicable.

< acl_item >

Stores command that are part of a IOS-XE configuration section. For example all configuration lines under an interface. It should not store configurations for configs that already have a separate item. For example BGP has a router item and should not also be stored in a acl_item.

Extends: oval-sc:ItemType

Child Elements

Table 536: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	Element with the name of the ACL.
ip_version	iosxe-sc:EntityItemAccessListIPVersionType (0..1)	Element with the IP version of the ACL.
use	iosxe-sc:EntityItemAccessListUseType (0..1)	Element with the feature where the ACL is used. If the same ACL is applied in more than one feature (i.e interface and crypto map), multiple items needs to be created.
used_in	oval-sc:EntityItemStringType (0..1)	Element with the name of where the ACL is used. For example if use is 'INTERFACE', use_in will be the name of the interface. If the same ACL is applied in more than one feature (i.e interface and crypto map), multiple items needs to be created.
interface_direction	iosxe-sc:EntityItemAccessListInterfaceDirectionType (0..1)	Element with the direction the ACL is applied on an interface.
acl_config_lines	oval-sc:EntityItemStringType (0..1)	Element with the value returned with all config lines of the ACL.
config_line	oval-sc:EntityItemStringType (0..unbounded)	Element with the value returned with one ACL config line at a time.

< snmphost_item >

Stores information about the SNMP host configuration in IOS. That information includes the host, the community or user strings, the SNMP version, the snmp security (if the SNMP version is SNMPv3) and the SNMP traps.

Extends: oval-sc:ItemType

Child Elements

Table 537: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
host	oval-sc:EntityItemStringType (0..1)	Element with the SNMP host address or hostname.
community_or_user	oval-sc:EntityItemStringType (0..1)	Element with the community string or SNMPv3 user configured for the host.
version	iosxe-sc:EntityItemSNMPVersionStringType (0..1)	Element with the SNMP version.
snmpv3_sec_level	iosxe-sc:EntityItemSNMPSecLevelStringType (0..1)	Element with the SNMPv3 security configure for the host.
traps	oval-sc:EntityItemStringType (0..1)	Element with the SNMP traps configured.

< snmpcommunity_item >

Stores information about an SNMP community configuration in IOS. That information includes the community name, the view (if it applies) name, the read-write mode and the ACLs names applied.

Extends: oval-sc:ItemType

Child Elements

Table 538: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	Element with the SNMP community name.
view	oval-sc:EntityItemStringType (0..1)	Element with the view that restricts the OIDs of this community.
mode	iosxe-sc:EntityItemSNMPModeStringType (0..1)	Element with the read-write privileges of the community.
ipv4_acl	oval-sc:EntityItemStringType (0..1)	Element with the IPv4 ACL name applied to the community.
ipv6_acl	oval-sc:EntityItemStringType (0..1)	Element with the IPv6 ACL name applied to the community

< snmpuser_item >

Stores information about an SNMP user configuration in IOS. That information includes the user name, the SNMP group he belongs to, the SNMP version, the IPv4 or IPv6 ACL it is applied to, the Security Level and the Authentication type that apply to the user (for SNMPv3).

Extends: oval-sc:ItemType

Child Elements

Table 539: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	Element with the SNMP user name.
group	oval-sc:EntityItemStringType (0..1)	Element with the SNMP group the user belongs to.
version	iosxe-sc:EntityItemSNMPVersionStringType (0..1)	Element with the SNMP version of the user.
ipv4_acl	oval-sc:EntityItemStringType (0..1)	Element with the IPv4 ACL name applied to the user.
ipv6_acl	oval-sc:EntityItemStringType (0..1)	Element with the IPv6 ACL name applied to the user.
priv	iosxe-sc:EntityItemSNMPPrivStringType (0..1)	Element with the SNMP encryption type for the user (for SNMPv3).
auth	iosxe-sc:EntityItemSNMPAuthStringType (0..1)	Element with the SNMP authentication type for the user (for SNMPv3).

< snmpgroup_item >

Stores information about an SNMP group configuration in IOS. That information includes the group name, the SNMP version, the IPv4 or IPv6 ACL it is applied toand the read, write and/or notify views applied to the group.

Extends: oval-sc:ItemType

Child Elements

Table 540: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	Element with the SNMP group name.
version	iosxe-sc:EntityItemSNMPVersionStringType (0..1)	Element with the SNMP version of the group.
snmpv3_sec_level	iosxe-sc:EntityItemSNMPSecLevelStringType (0..1)	Element with the SNMPv3 security configure for the group.
ipv4_acl	oval-sc:EntityItemStringType (0..1)	Element with the IPv4 ACL name applied to the group.
ipv6_acl	oval-sc:EntityItemStringType (0..1)	Element with the IPv6 ACL name applied to the group.
read_view	oval-sc:EntityItemStringType (0..1)	Element with the SNMP read view applied to the group.
write_view	oval-sc:EntityItemStringType (0..1)	Element with the SNMP write view applied to the group.
notify_view	oval-sc:EntityItemStringType (0..1)	Element with the SNMP notify view applied to the group.

< snmpview_item >

Stores information about an SNMP view configuration in IOS. That information includes the view name, the mib_family that the view uses and the included or excluded option of the mib family in the view.

Extends: oval-sc:ItemType

Child Elements

Table 541: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	Element with the SNMP view name.
mib_family	oval-sc:EntityItemStringType (0..1)	Element with the SNMP MIB family of the view.
include	oval-sc:EntityItemBoolType (0..1)	Element that is true if the included option is used in the view.

== EntityItemTrunkEncapType ==

The EntityItemTrunkEncapType complex type restricts a string value to a specific set of values: DOT1Q, ISL, NEGOTIATE. These values describe the interface trunk encapsulation types on an interfaces in IOS. The empty string is also allowed to support empty element associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 542: Enumeration Values

Value	Description
DOT1Q	(No Description)
ISL	(No Description)
NEGOTIATE	(No Description)
	The empty string value is permitted here to allow for empty elements associated with error conditions.

== EntityItemSwitchportModeType ==

The EntityObjectRoutingProtocolType complex type restricts a string value to a specific set of values: DYNAMIC, TRUNK, ACCESS. These values describe the interface switchport mode types in IOS. The empty string is also allowed to support empty element associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 543: Enumeration Values

Value	Description
DYNAMIC	(No Description)
TRUNK	(No Description)
ACCESS	(No Description)
	The empty string value is permitted here to allow for empty elements associated with error conditions.

== EntityItemRoutingProtocolType ==

The EntityItemRoutingProtocolType complex type restricts a string value to a specific set of values: EIGRP, OSPF, BGP, RIP, RIPV2, ISIS. These values describe the routing protocol used in a Cisco IOS-XE configuration. The empty string is also allowed to support empty element associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 544: Enumeration Values

Value	Description
EIGRP	(No Description)
OSPF	(No Description)
BGP	(No Description)
RIP	(No Description)
RIPV2	(No Description)
ISIS	(No Description)
	The empty string value is permitted here to allow for empty elements associated with error conditions.

== EntityItemRoutingAuthTypeStringType ==

The EntityItemRoutingAuthTypeStringType complex type restricts a string value to a specific set of values: CLEAR-TEXT, MESSAGE_DIGEST. These values describe the routing protocol authentication types used in a Cisco IOS-XE configuration. The empty string is also allowed to support empty element associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 545: Enumeration Values

Value	Description
CLEARTEXT	(No Description)
MESSAGE_DIGEST	(No Description)
NULL (Deprecated)	**Deprecated As Of Version:** 5.11.2:1.0 **Reason:** The NULL authentication area type is never declared in an interface ip ospf command context. **Comment:** This RoutingAuthTypeStringType enumeration value has been deprecated and may be removed in a future version of the language.
	The empty string value is permitted here to allow for empty elements associated with error conditions.

== EntityItemSNMPVersionStringType ==

The EntityItemSNMPVersionStringType complex type restricts a string value to a specific set of values: 1, 2c, 3. These values describe the SNMP version in a Cisco IOS-XE configuration. The empty string is also allowed to support empty element associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 546: Enumeration Values

Value	Description
1	(No Description)
2C	(No Description)
3	(No Description)
	The empty string value is permitted here to allow for empty elements associated with error conditions.

== EntityItemSNMPSecLevelStringType ==

The EntityItemSNMPVersionStringType complex type restricts a string value to a specific set of values: PRIV, AUTH, NO_AUTH. These values describe the SNMP security level (encryption, Authentication, None) in a Cisco IOS-XE SNMPv3 related configurations. The empty string is also allowed to support empty element associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 547: Enumeration Values

Value	Description
PRIV	(No Description)
AUTH	(No Description)
NO_AUTH	(No Description)
	The empty string value is permitted here to allow for empty elements associated with error conditions.

== EntityItemSNMPModeStringType ==

The EntityItemSNMPModeStringType complex type restricts a string value to a specific set of values: RO, RW. These values describe the SNMP mode (read-only, read-write) in a Cisco IOS-XE SNMPv3 related configurations. The empty string is also allowed to support empty element associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 548: Enumeration Values

Value	Description
RO	(No Description)
RW	(No Description)
	The empty string value is permitted here to allow for empty elements associated with error conditions.

== EntityItemSNMPAuthStringType ==

The EntityItemSNMPAuthStringType complex type restricts a string value to a specific set of values: MD5, SHA. These values describe the authentication algorithm in a Cisco IOS-XE SNMPv3 related configurations. The empty string is also allowed to support empty element associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 549: Enumeration Values

Value	Description
MD5	(No Description)
SHA	(No Description)
	The empty string value is permitted here to allow for empty elements associated with error conditions.

== EntityItemSNMPPrivStringType ==

The EntityItemSNMPPrivStringType complex type restricts a string value to a specific set of values: DES, 3DES, AES. These values describe the encryption algorithm in a Cisco IOS-XE SNMPv3 related configurations. The empty string is also allowed to support empty element associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 550: Enumeration Values

Value	Description
DES	(No Description)
3DES	(No Description)
AES	(No Description)
	The empty string value is permitted here to allow for empty elements associated with error conditions.

== EntityItemAccessListIPVersionType ==

The EntityItemRoutingProtocolType complex type restricts a string value to a specific set of values: IPV4, IPV6. These values describe if an ACL is for IPv4 or IPv6 in a Cisco IOS-XE configuration. The empty string is also allowed to support empty element associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 551: Enumeration Values

Value	Description
IPV4	(No Description)
IPV6	(No Description)
	The empty string value is permitted here to allow for empty elements associated with error conditions.

== EntityItemAccessListUseType ==

The EntityItemAccessListUseType complex type restricts a string value to a specific set of values: INTERFACE, CRYPTO_MAP_MATCH, CLASS_MAP_MATCH, ROUTE_MAP_MATCH, IGMP_FILTER, VTY. These values describe the ACL use in a Cisco IOS-XE configuration. The empty string is also allowed to support empty element associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 552: Enumeration Values

Value	Description
INTERFACE	(No Description)
CRYPTO_MAP_MATCH	(No Description)
CLASS_MAP_MATCH	(No Description)
ROUTE_MAP_MATCH	(No Description)
IGMP_FILTER	(No Description)
VTY	(No Description)
NONE (Deprecated)	**Deprecated As Of Version:** 5.11.2:1.0 **Reason:** The EntityStateSimpleBaseType check_existence attribute serves the same purpose as this enumeration value. **Comment:** This AccessListUseType enumeration value has been deprecated and may be removed in a future version of the language.
	The empty string value is permitted here to allow for empty elements associated with error conditions.

== EntityItemAccessListInterfaceDirectionType ==

The EntityItemAccessListInterfaceDirectionType complex type restricts a string value to a specific set of values: IN, OUT. These values describe the inbound or outbound ACL direction on an interface in a Cisco IOS-XE configuration. The empty string is also allowed to support empty element associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 553: Enumeration Values

Value	Description
IN	(No Description)
OUT	(No Description)
	The empty string value is permitted here to allow for empty elements associated with error conditions.

Open Vulnerability and Assessment Language: PixOS Definition

- Schema: PixOS Definition
- Version: 5.11.1:1.1
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the PIX specific tests found in Open Vulnerability and Assessment Language (OVAL). Each test is an extension of the standard test element defined in the

Core Definition Schema. Through extension, each test inherits a set of elements and attributes that are shared amongst all OVAL tests. Each test is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core Definition Schema is not outlined here.

This schema was originally developed by Yuzheng Zhou and Eric Grey at Hewlett-Packard. The OVAL Schema is maintained by the 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.cisecurity.org.

Test Listing

- *< line_test >*
- *< version_test >*

< line_test >

The line_test is used to check the properties of specific output lines from a SHOW command, such as SHOW RUNNING-CONFIG. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a line_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 554: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< line_object >

The line_object element is used by a line_test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A line object consists of a show_subcommand entity that is the name of a SHOW sub-command to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 555: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
show_subcommand	oval-def:EntityObjectStringType (1..1)	The name of a SHOW sub-command.
oval-def:filter	n/a (0..unbounded)	

< line_state >

The line_state element defines the different information that can be used to evaluate the result of a specific SHOW sub-command. This includes the name of ths sub-command and the corresponding config line. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 556: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
show_subcommand	oval-def:EntityStateStringType (0..1)	The name of the SHOW sub-command.
config_line	oval-def:EntityStateStringType (0..1)	The value returned from by the specified SHOW sub-command.

< version_test >

The version test is used to check the version of the PIX operating system. It is based off of the SHOW VERSION command. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a version_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 557: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< version_object >

The version_object element is used by a version test to define the different version information associated with a PIX system. There is actually only one object relating to version and this is the system as a whole. Therefore, there are no child entities defined. Any OVAL Test written to check version will reference the same version_object which is basically an empty object element.

Extends: oval-def:ObjectType

< version_state >

The version_state element defines the version information held within a Cisco PIX software release. The pix_release element specifies the whole PIX version information. The pix_major_release, pix_minor_release and pix_build elements specify seperated parts of PIX software version information. For instance, if the PIX version is 7.1(2.3)49, then pix_release is 7.1(2.3)49, pix_major_release is 7.1, pix_minor_release is 2.3 and pix_build is 49. See the SHOW VERSION command within PIX for more information.

Extends: oval-def:StateType

Child Elements

Table 558: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
pix_release	oval-def:EntityStateStringType (0..1)	The pix_release element specifies the whole PIX version information.
pix_major_release	oval-def:EntityStateVersionType (0..1)	The pix_major_release is the dotted version that starts a version string. For example the pix_release 7.1(2.3)49 has a pix_major_release of 7.1.
pix_minor_release	oval-def:EntityStateVersionType (0..1)	The pix_minor_release is the dotted version that starts a version string. For example the pix_release 7.1(2.3)49 has a pix_minor_release of 2.3.
pix_build	oval-def:EntityStateIntType (0..1)	The pix_build is an integer. For example the pix_release 7.1(2.3)49 has a pix_build of 49.

Open Vulnerability and Assessment Language: PixOS System Characteristics

- Schema: PixOS System Characteristics
- Version: 5.11.1:1.1
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the Cisco PIX (Private Internet Exchange) specific system characteristic items found in Open Vulnerability and Assessment Language (OVAL). Each item is an extension of the standard test element defined in the Core Definition Schema. Through extension, each test inherits a set of elements and attributes that are shared amongst all OVAL tests. Each test is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core Definition Schema is not outlined here.

This schema was originally developed by Yuzheng Zhou and Eric Grey at Hewlett-Packard. The OVAL Schema is maintained by the 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.cisecurity.org.

Item Listing

- *< line_item >*
- *< version_item >*

< line_item >

Stores the properties of specific lines in the PIX config file.

Extends: oval-sc:ItemType

Child Elements

Table 559: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
show_subcommand	oval-sc:EntityItemStringType (0..1)	The name of the SHOW sub-command.
config_line	oval-sc:EntityItemStringType (0..1)	The value returned from by the specified SHOW sub-command.

< version_item >

Stores results from SHOW VERSION command.

Extends: oval-sc:ItemType

Child Elements

Table 560: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
pix_release	oval-sc:EntityItemStringType (0..1)	
pix_major_release	oval-sc:EntityItemVersionType (0..1)	
pix_minor_release	oval-sc:EntityItemVersionType (0..1)	
pix_build	oval-sc:EntityItemIntType (0..1)	

Open Vulnerability and Assessment Language: Junos Definition

- Schema: Junos Definition
- Version: 5.11.1:1.1
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the Junos-specific tests found in Open Vulnerability and Assessment Language (OVAL). Each test is an extension of the standard test element defined in the Core Definition Schema. Through extension, each test inherits a set of elements and attributes that are shared amongst all OVAL tests. Each test is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core Definition Schema is not outlined here.

This schema was originally developed by David Solin at jOVAL.org. The OVAL Schema is maintained by the 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.cisecurity.org.

Test Listing

- *< xml_config_test >*
- *< show_test >*
- *< version_test >*
- *< xml_show_test >*

< xml_config_test >

Extends: oval-def:TestType

Child Elements

Table 561: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< xml_config_object >

The xml_config_object element is used by an XML config test to define the object to be evaluated. For the most part this object checks for existence and is used without a state comparision. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 562: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
xpath	oval-def:EntityObjectStringType (1..1)	An XPATH 1.0 expression that should be evaluated against the XML configuration file. 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 given xpath would be impossible.
oval-def:filter	n/a (0..unbounded)	

< xml_config_state >

The xml_config_state element defines the different information that can be used to evaluate the result of an XPATH query against the XML configuration file. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 563: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
xpath	oval-def:EntityStateStringType (0..1)	An XPATH 1.0 expression that was evaluated against the XML config file.
value_of	oval-def:EntityStateAnySimpleType (0..1)	The result of the evaluation of the XPATH expression against the XML config file.

< show_test >

The show test is used to check the properties of specific output lines from a SHOW command, such as "show configuration". It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a show_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 564: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< show_object >

The show_object element is used by a show test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 565: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
subcommand	oval-def:EntityObjectStringType (1..1)	The name of a SHOW sub-command to be tested.
oval-def:filter	n/a (0..unbounded)	

< show_state >

The show_state element defines the different information that can be used to evaluate the result of a specific SHOW sub-command. This includes the name of the sub-command and the corresponding config output. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 566: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
subcommand	oval-def:EntityStateStringType (0..1)	The name of the SHOW sub-command.
value	oval-def:EntityStateStringType (0..1)	The value returned from by the specified SHOW sub-command. This may consist of multiple lines of information, whose raw form will be captured by the item.

< version_test >

The version_test is used to check the version of components of the JunOS operating system. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a version_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 567: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< version_object >

The version_object element is used by a version_test to define the different version information associated with a JunOS system.

Extends: oval-def:ObjectType

Child Elements

Table 568: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
component	oval-def:EntityObjectStringType (1..1)	The name of the JunOS component whose version should be retrieved.
oval-def:filter	n/a (0..unbounded)	

< version_state >

The version_state element defines the version information held by a JunOS component.

Extends: oval-def:StateType

Child Elements

Table 569: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
component	oval-def:EntityStateStringType (0..1)	The name of the JunOS component whose version should be retrieved.
raw_value	oval-def:EntityStateStringType (0..1)	The raw release version string for the component, e.g., 12.2R6.1 or 12.1X44-D10.4.
major	oval-def:EntityStateIntType (0..1)	The part of the release version of the component corresponding to the year in which the release occurred. For example, the major value for 12.2R6.1 would be ‘12’.
minor	oval-def:EntityStateIntType (0..1)	The part of the release version of the component corresponding to the quarter in which the release occurred. For example, the minor value for 12.2R6.1 would be ‘2’.
type	junos-def:EntityStateJunosReleaseTypeType (0..1)	The release type embedded in the version of the component. For example, the type value for 12.2R6.1 is ‘R’.
build	oval-def:EntityStateIntType (0..1)	The build number of the component’s version. For example, the revision for 12.2R6.1 has a build number of ‘6’; 12.1X44-D10.4 has a build number of ‘44’.
maintenance_release	oval-def:EntityStateIntType (0..1)	A maintenance_release value can appear in an R-type service release or an X-type release (where it takes the value of the D-number). For example, version 14.2R3-S4.5 has a maintenance_release of ‘4’. For version 10.4S4.2, the maintenance_release entity would have a status of ‘does not exist’. For version 12.1X44-D10.4, the maintenance_release entity value would be ‘10’.
spin	oval-def:EntityStateIntType (0..1)	The spin number of the component. For example, 12.2R6.1 has a spin value of ‘1’; 12.1X44-D10.4 has a spin value of ‘4’.
build_date	oval-def:EntityStateIntType (0..1)	The build date of the component, specified in milliseconds since the Epoch (midnight, January 1, 1970 GMT).

< xml_show_test >

The XML show test is used to check the properties of specific output from an XML SHOW command, such as “show configuration | display xml”. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a xml_show_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 570: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< xml_show_object >

The xml_show_object element is used by an XML show test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 571: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
sub-command	oval-def:EntityObjectStringType (1..1)	The name of a SHOW sub-command to be tested.
xpath	oval-def:EntityObjectStringType (1..1)	An XPATH 1.0 expression that should be evaluated against the XML data resulting from the XML SHOW sub-command. 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 given xpath would be impossible.
oval-def:filter	n/a (0..unbounded)	

< xml_show_state >

The xml_show_state element defines the different information that can be used to evaluate the result of a specific XML SHOW sub-command. This includes the name of the sub-command, the XPATH and the corresponding XPATH query result. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 572: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
subcommand	oval-def:EntityStateStringType (0..1)	The name of a SHOW sub-command to be tested.
xpath	oval-def:EntityStateStringType (0..1)	An XPATH 1.0 expression that should be evaluated against the XML data resulting from the XML show subcommand.
value_of	oval-def:EntityStateAnySimpleType (0..1)	The result of the evaluation of the XPATH expression against the XML data returned from the XML show subcommand.

== EntityStateJunosReleaseTypeType ==

The EntityStateJunosReleaseTypeType complex type defines the different values that are valid for the release_type entity of a system_metric state. These values describe the release type specified in the raw version string.

Restricts: oval-def:EntityStateStringType

Table 573: Enumeration Values

Value	Description
R	Indicates a normal release.
I	Indicates an internal release.
F	Indicates a feature release.
S	Indicates a service release.
B	Indicates a beta release.
X	Indicates an exception release (e.g., every release of the SRX branch so far).
	The empty string value is permitted here to allow for empty elements associated with variable references.

Open Vulnerability and Assessment Language: Junos System Characteristics

- Schema: Junos System Characteristics
- Version: 5.11.1:1.1
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the Junos-specific system characteristic items found in Open Vulnerability and Assessment Language (OVAL). Each item is an extension of the standard item element defined in the Core System Characteristic Schema. Through extension, each item inherits a set of elements and attributes that are shared amongst all OVAL Items. Each item is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core System Characteristic Schema is not outlined here.

This schema was originally developed by David Solin at jOVAL.org. The OVAL Schema is maintained by the 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.cisecurity.org.

Item Listing

- *< xml_config_item >*
- *< show_item >*
- *< version_item >*
- *< xml_show_item >*

< xml_config_item >

Stores information about the existence of a particular XPATH query result from the JunOS XML config file.

Extends: oval-sc:ItemType

Child Elements

Table 574: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
xpath	oval-sc:EntityItemStringType (0..1)	An XPATH 1.0 expression that was evaluated against the XML config file.
value_of	oval-sc:EntityItemAnySimpleType (0..unbounded)	The result of the evaluation of the XPATH expression against the XML config file.

< show_item >

Stores the resulting configuration data provided by the execution of a specific show command.

Extends: oval-sc:ItemType

Child Elements

Table 575: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
subcommand	oval-sc:EntityItemStringType (0..1)	The name of the SHOW sub-command.
value	oval-sc:EntityItemStringType (0..1)	The value returned from by the specified SHOW sub-command. This may consist of multiple lines of information.

< version_item >

The version_item holds information about the version of a particular component of the JunOS operating system. It extends the standard ItemType as defined in the oval-system-characteristics schema and one should refer to the ItemType description for more information.

Extends: oval-sc:ItemType

Child Elements

Table 576: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
component	oval-sc:EntityItemStringType (0..1)	The name of the JunOS component whose version should be retrieved.
raw_value	oval-sc:EntityItemStringType (0..1)	The raw release version string for the component, e.g., 12.2R6.1 or 12.1X44-D10.4.
major	oval-sc:EntityItemIntType (0..1)	The part of the release version of the component corresponding to the year in which the release occurred. For example, the major value for 12.2R6.1 would be '12'.
minor	oval-sc:EntityItemIntType (0..1)	The part of the release version of the component corresponding to the quarter in which the release occurred. For example, the minor value for 12.2R6.1 would be '2'.
type	junos-sc:EntityItemJunosReleaseTypeType (0..1)	The release type embedded in the version of the component. For example, the type value for 12.2R6.1 is 'R'.
build	oval-sc:EntityItemIntType (0..1)	The build number of the component's version. For example, the revision for 12.2R6.1 has a build number of '6'; 12.1X44-D10.4 has a build number of '44'.
maintenance_release	oval-sc:EntityItemIntType (0..1)	A maintenance_release value can appear in an R-type service release or an X-type release (where it takes the value of the D-number). For example, version 14.2R3-S4.5 has a maintenance_release of '4'. For version 10.4S4.2, the maintenance_release entity would have a status of 'does not exist'. For version 12.1X44-D10.4, the maintenance_release entity value would be '10'.
spin	oval-sc:EntityItemIntType (0..1)	The spin number of the component. For example, 12.2R6.1 has a spin value of '1'; 12.1X44-D10.4 has a spin value of '4'.
build_date	oval-sc:EntityItemIntType (0..1)	The build date of the component, specified in milliseconds since the Epoch (midnight, January 1, 1970 GMT).

< xml_show_item >

Stores the result of the application of an XPATH query applied to the JunOS configuration data provided by the execution of a specific show command, which has been piped to "display xml".

Extends: oval-sc:ItemType

Child Elements

Table 577: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
subcommand	oval-sc:EntityItemStringType (0..1)	The name of a SHOW sub-command to be tested.
xpath	oval-sc:EntityItemStringType (0..1)	An XPATH 1.0 expression that should be evaluated against the XML data resulting from the XML show subcommand.
value_of	oval-sc:EntityItemAnySimpleType (0..unbounded)	The result of the evaluation of the XPATH expression against the XML data returned from the XML show subcommand.

== EntityItemJunosReleaseTypeType ==

The EntityItemJunosReleaseTypeType complex type defines the different values that are valid for the release_type entity of a system_metric state. These values describe the release type specified in the raw version string.

Restricts: oval-sc:EntityItemStringType

Table 578: Enumeration Values

Value	Description
R	Indicates a normal release.
I	Indicates an internal release.
F	Indicates a feature release.
S	Indicates a service release.
B	Indicates a beta release.
X	Indicates an exception release (e.g., every release of the SRX branch so far).
	The empty string value is permitted here to allow for empty elements associated with error conditions.

Open Vulnerability and Assessment Language: NETCONF Definitions

- Schema: NETCONF Definitions
- Version: 5.11.1:1.1
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the NETCONF (RFC 6241) protocol-specific tests found in Open Vulnerability and Assessment Language (OVAL). Each test is an extension of the standard test element defined in the Core Definition Schema. Through extension, each test inherits a set of elements and attributes that are shared amongst all OVAL tests. Each test is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core Definition Schema is not outlined here

This schema was originally developed by David Solin at jOVAL.org. The OVAL Schema is maintained by the 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.cisecurity.org.

Test Listing

- *< config_test >*

< config_test >

The config_test is used to check the properties of the XML output from a GET-CONFIG command, for the running configuration. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a config_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 579: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< config_object >

The config_object element is used by a config_test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A config_object consists of an xpath entity that contains an XPATH 1.0 query to perform on the NETCONF get-config response XML data. The response data is assumed to consist of a <data> entity in the urn:ietf:params:xml:ns:netconf:base:1.0 XML namespace, with arbitrary (i.e., vendor-specific) child nodes.

Extends: oval-def:ObjectType

Child Elements

Table 580: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
xpath	oval-def:EntityObjectStringType (1..1)	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 given xpath would be impossible.
oval-def:filter	n/a (0..unbounded)	

< config_state >

The config_state element defines the different information that can be used to evaluate the result of a specific config xpath evaluation. This includes the xpath used and the value of this xpath.

Extends: oval-def:StateType

Child Elements

Table 581: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
xpath	oval-def:EntityStateStringType (0..1)	Specifies an Xpath expression describing the text node(s) or attribute(s) to look at.
value_of	oval-def:EntityStateAnySimpleType (0..1)	The value_of element checks the value(s) of the text node(s) or attribute(s) found.

Open Vulnerability and Assessment Language: NETCONF System Characteristics

- Schema: NETCONF System Characteristics
- Version: 5.11.1:1.1
- Release Date: 11/30/2016 09:00:00 AM

This document outlines the items of the OVAL System Characteristics XML schema that are composed of NETCONF (RFC 6241) protocol-specific tests. Each item is an extention of a basic System Characteristics item defined in the core System Characteristics XML schema.

This schema was originally developed by David Solin at jOVAL.org. The OVAL Schema is maintained by the 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.cisecurity.org.

Item Listing

- *< config_item >*

< config_item >

This item stores results from checking the contents of an xml configuration.

Extends: oval-sc:ItemType

Child Elements

Table 582: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
xpath	oval-sc:EntityItemStringType (0..1)	Specifies an Xpath expression describing the text node(s) or attribute(s) to look at.
value_of	oval-sc:EntityItemAnySimpleType (0..unbounded)	The value_of element checks the value(s) of the text node(s) or attribute(s) found. How this is used is entirely controlled by operator attributes.

Open Vulnerability and Assessment Language: Windows Definition

- Schema: Windows Definition
- Version: 5.11.1:1.4
- Release Date: 01/09/2017 10:00:00 PM

The following is a description of the elements, types, and attributes that compose the Windows specific tests found in Open Vulnerability and Assessment Language (OVAL). Each test is an extension of the standard test element defined in the Core Definition Schema. Through extension, each test inherits a set of elements and attributes that are shared amongst all OVAL tests. Each test is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core Definition Schema is not outlined here.

The OVAL Schema is maintained by the 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.cisecurity.org.

Test Listing

- *< accesstoken_test > (Deprecated)* (Deprecated)
- *< activedirectory_test >*
- *< activedirectory57_test > (Deprecated)* (Deprecated)
- *< auditeventpolicy_test >*
- *< auditeventpolicysubcategories_test >*
- *< cmdlet_test >*
- *< dnscache_test >*
- *< file_test >*
- *< fileauditedpermissions53_test >*
- *< fileauditedpermissions_test > (Deprecated)* (Deprecated)
- *< fileeffectiverights53_test >*
- *< fileeffectiverights_test > (Deprecated)* (Deprecated)
- *< group_test > (Deprecated)* (Deprecated)
- *< group_sid_test >*
- *< interface_test >*
- *< junction_test >*
- *< license_test >*
- *< lockoutpolicy_test >*
- *< metabase_test >*
- *< ntuser_test >*
- *< passwordpolicy_test >*
- *< peheader_test >*
- *< port_test >*
- *< printereffectiverights_test >*
- *< process_test > (Deprecated)* (Deprecated)
- *< process58_test >*
- *< registry_test >*
- *< regkeyauditedpermissions53_test >*
- *< regkeyauditedpermissions_test > (Deprecated)* (Deprecated)
- *< regkeyeffectiverights53_test >*
- *< regkeyeffectiverights_test > (Deprecated)* (Deprecated)
- *< service_test >*
- *< serviceeffectiverights_test >*
- *< sharedresource_test >*
- *< sharedresourceauditedpermissions_test >*

- *< sharedresourceeffectiverights_test >*
- *< sid_test >*
- *< sid_sid_test >*
- *< systemmetric_test >*
- *< uac_test >*
- *< user_test > (Deprecated)* (Deprecated)
- *< user_sid55_test >*
- *< user_sid_test > (Deprecated)* (Deprecated)
- *< userright_test >*
- *< volume_test >*
- *< wmi_test > (Deprecated)* (Deprecated)
- *< wmi57_test >*
- *< wuaupdatesearcher_test >*

< accesstoken_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.11
- Reason: Replaced by the userright_test. This accesstoken_test suffers from scalability issues when run on a domain controller and should not be used. See the userright_test.
- Comment: This test has been deprecated and will be removed in version 6.0 of the language.

The accesstoken_test is used to check the properties of a Windows access token as well as individual privileges and rights associated with it. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an accesstoken_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 583: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< accesstoken_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.11

- Reason: Replaced by the userright_object. The accesstoken_test suffers from scalability issues when run on a domain controller and should not be used. See the userright_object.
- Comment: This object has been deprecated and will be removed in version 6.0 of the language.

The accesstoken_object element is used by an access token test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An accesstoken_object consists of a single security principle that identifies user, group, or computer account that is associated with the token.

Extends: oval-def:ObjectType

Child Elements

Table 584: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	win-def:AccesstokenBehaviors (0..1)	
security_principle	oval-def:EntityObjectStringType (1..1)	The security_principle element defines the access token being specified. Security principles include users or groups with either local or domain accounts, and computer accounts created when a computer joins a domain. In Windows, security principles are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. User rights and permissions to access objects such as Active Directory objects, files, and registry settings are assigned to security principles. In a domain environment, security principles should be identified in the form: “domaintrustee name”. For local security principles use: “computer nametrustee name”. For built-in accounts on the system, use the trustee name without a domain. If an operation other than equals is used to identify matching trustees (i.e. not equal, or a pattern match) then the resulting matches shall be limited to only the trustees referenced in the Local Security Authority database. The scope is limited here to avoid unnecessarily resource intensive searches for trustees. Note that the larger scope of all known trustees may be obtained through the use of variables.
oval-def:filter	n/a (0..unbounded)	

< accesstoken_state > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.11
- Reason: Replaced by the userright_state. The accesstoken_test suffers from scalability issues when run on a domain controller and should not be used. See the userright_state.
- Comment: This state has been deprecated and will be removed in version 6.0 of the language.

The accesstoken_state element defines the different information that can be used to evaluate the specified access tokens. This includes the multitude of user rights and permissions that can be granted. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
security_principle	oval-def:EntityStateStringType (0..1)	The security_principle element identifies an access toke
seassignprimarytokenprivilege	oval-def:EntityStateBoolType (0..1)	If the seassignprimarytokenprivilege privilege is enable
seauditprivilege	oval-def:EntityStateBoolType (0..1)	If the seauditprivilege privilege is enabled, it allows a p
sebackupprivilege	oval-def:EntityStateBoolType (0..1)	If the sebackupprivilege privilege is enabled, it allows t
sechangenotifyprivilege	oval-def:EntityStateBoolType (0..1)	If the sechangenotifyprivilege privilege is enabled, it all
secreateglobalprivilege	oval-def:EntityStateBoolType (0..1)	If the secreateglobalprivilege privilege is enabled, it allo
secreatepagefileprivilege	oval-def:EntityStateBoolType (0..1)	If the secreatepagefileprivilege privilege is enabled, it al
secreatepermanentprivilege	oval-def:EntityStateBoolType (0..1)	If the secreatepermanentprivilege privilege is enabled, it
secreatesymboliclinkprivilege	oval-def:EntityStateBoolType (0..1)	If the secreatesymboliclinkprivilege privilege is enabled
secreatetokenprivilege	oval-def:EntityStateBoolType (0..1)	If the secreatetokenprivilege privilege is enabled, it allo
sedebugprivilege	oval-def:EntityStateBoolType (0..1)	If the sedebugprivilege privilege is enabled, it allows the
seenabledelegationprivilege	oval-def:EntityStateBoolType (0..1)	If the seenabledelegationprivilege privilege is enabled, i
seimpersonateprivilege	oval-def:EntityStateBoolType (0..1)	If the seimpersonateprivilege privilege is enabled, it allo
seincreasebasepriorityprivilege	oval-def:EntityStateBoolType (0..1)	If the seincreasebasepriorityprivilege privilege is enable
seincreasequotaprivilege	oval-def:EntityStateBoolType (0..1)	If the seincreasequotaprivilege privilege is enabled, it al
seincreaseworkingsetprivilege	oval-def:EntityStateBoolType (0..1)	If the seincreaseworkingsetprivilege privilege is enabled
seloaddriverprivilege	oval-def:EntityStateBoolType (0..1)	If the seloaddriverprivilege privilege is enabled, it allow
selockmemoryprivilege	oval-def:EntityStateBoolType (0..1)	If the selockmemoryprivilege privilege is enabled, it allo
semachineaccountprivilege	oval-def:EntityStateBoolType (0..1)	If the semachineaccountprivilege privilege is enabled, it
semanagevolumeprivilege	oval-def:EntityStateBoolType (0..1)	If the semanagevolumeprivilege privilege is enabled, it a
seprofilesingleprocessprivilege	oval-def:EntityStateBoolType (0..1)	If the seprofilesingleprocessprivilege privilege is enable
serelabelprivilege	oval-def:EntityStateBoolType (0..1)	If the serelabelprivilege privilege is enabled, it allows a
seremoteshutdownprivilege	oval-def:EntityStateBoolType (0..1)	If the seremoteshutdownprivilege privilege is enabled, it
serestoreprivilege	oval-def:EntityStateBoolType (0..1)	If the serestoreprivilege privilege is enabled, it allows a
sesecurityprivilege	oval-def:EntityStateBoolType (0..1)	If the sesecurityprivilege privilege is enabled, it allows a
seshutdownprivilege	oval-def:EntityStateBoolType (0..1)	If the seshutdownprivilege privilege is enabled, it allows
sesyncagentprivilege	oval-def:EntityStateBoolType (0..1)	If the sesyncagentprivilege privilege is enabled, it allow
sesystemenvironmentprivilege	oval-def:EntityStateBoolType (0..1)	If the sesystemenvironmentprivilege privilege is enabled
sesystemprofileprivilege	oval-def:EntityStateBoolType (0..1)	If the sesystemprofileprivilege privilege is enabled, it al
sesystemtimeprivilege	oval-def:EntityStateBoolType (0..1)	If the sesystemtimeprivilege privilege is enabled, it allo
setakeownershipprivilege	oval-def:EntityStateBoolType (0..1)	If the setakeownershipprivilege privilege is enabled, it a
setcbprivilege	oval-def:EntityStateBoolType (0..1)	If the setcbprivilege privilege is enabled, it allows a pro
setimezoneprivilege	oval-def:EntityStateBoolType (0..1)	If the setimezoneprivilege privilege is enabled, it allows
seundockprivilege	oval-def:EntityStateBoolType (0..1)	If the seundockprivilege privilege is enabled, it allows t
seunsolicitedinputprivilege	oval-def:EntityStateBoolType (0..1)	If the seunsolicitedinputprivilege privilege is enabled, it
sebatchlogonright	oval-def:EntityStateBoolType (0..1)	If an account is assigned the sebatchlogonright right, it
seinteractivelogonright	oval-def:EntityStateBoolType (0..1)	If an account is assigned the seinteractivelogonright rig
senetworklogonright	oval-def:EntityStateBoolType (0..1)	If an account is assigned the senetworklogonright right,
seremoteinteractivelogonright	oval-def:EntityStateBoolType (0..1)	If an account is assigned the seremoteinteractivelogonri
seservicelogonright	oval-def:EntityStateBoolType (0..1)	If an account is assigned the seservicelogonright right, i

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
sedenybatchLogonright	oval-def:EntityStateBoolType (0..1)	If an account is assigned the sedenybatchLogonright rig
sedenyinteractivelogonright	oval-def:EntityStateBoolType (0..1)	If an account is assigned the sedenyinteractivelogonrigh
sedenynetworklogonright	oval-def:EntityStateBoolType (0..1)	If an account is assigned the sedenynetworklogonright r
sedenyremoteInteractivelogonright	oval-def:EntityStateBoolType (0..1)	If an account is assigned the sedenyremoteInteractivelog
sedenyservicelogonright	oval-def:EntityStateBoolType (0..1)	If an account is assigned the sedenyservicelogonright rig
setrustedcredmanaccessnameright	oval-def:EntityStateBoolType (0..1)	If an account is assigned this right, it can access the Cre

== AccesstokenBehaviors == (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.11
- Reason: Replaced by the userright_test. The AccesstokenBehaviors complex type is used by the accesstoken_test which suffers from scalability issues when run on a domain controller and should not be used. As a result, the AccesstokenBehaviors complex type is no longer needed. See the userright_test.
- Comment: This complex type has been deprecated and will be removed in version 6.0 of the language.

The AccesstokenBehaviors complex type defines a number of behaviors that allow a more detailed definition of the accesstoken_object being specified. 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.

Attributes

Table 586: Attributes

Attribute	Type	Desc.
include_group (Deprecated)	xsd:boolean (optional *default*='true')	If a group security principle is specified, this behavior specifies whether to include the group or not. For example, maybe you want to check the access tokens associated with every user within a group, but not the group itself. In this case, you would set the include_group behavior to 'false'. If the security_principle is not a group, then this behavior should be ignored.
resolve_group (Deprecated)	xsd:boolean (optional *default*='false')	The 'resolve_group' behavior defines whether an object set defined by a group SID should be resolved to return a set that contains all the user SIDs that are a member of that group. Note that all child groups should also be resolved and any valid domain users that are members of the group should also be included. The intent of this behavior is to end up with a list of all individual users from that system that make up the group once everything has been resolved.

< activedirectory_test >

The active directory test is used to check information about specific entries in active directory. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an activedirectory_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 587: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< activedirectory_object >

The activedirectory_object element is used by an active directory test to define those objects to evaluated based on a specified state. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An active directory object consists of three pieces of information, a naming context, a relative distinguished name, and an attribute. Each piece helps identify a specific active directory entry.

Extends: oval-def:ObjectType

Child Elements

Table 588: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
naming_context	win-def:EntityObjectNamingContextType (1..1)	Each object in active directory exists under a certain naming context (also known as a partition). A naming context is defined as a single object in the Directory Information Tree (DIT) along with every object in the tree subordinate to it. There are three default naming contexts in Active Directory: domain, configuration, and schema.
relative_dn	oval-def:EntityObjectStringType (1..1)	The relative_dn field is used to uniquely identify an object inside the specified naming context. It contains all the parts of the object's distinguished name except those outlined by the naming context. If the xsi:nil attribute is set to true, then the object being specified is the higher level naming context. In this case, the relative_dn element should not be collected or used in analysis. Setting xsi:nil equal to true is different than using a .* pattern match, which says to collect every relative dn under a given naming context.
attribute	oval-def:EntityObjectStringType (1..1)	Specifies a named value contained by the object. If the xsi:nil attribute is set to true, the attribute element should not be collected or used in analysis. Setting xsi:nil equal to true is different than using a .* pattern match, which says to collect every attribute under a given relative dn.

< activedirectory_state >

The activedirectory_state element defines the different information that can be used to evaluate the specified entries in active directory. An active directory test will reference a specific instance of this state that defines the exact settings that need to be evaluated. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 589: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
naming_context	win-def:EntityStateNamingContextType (0..1)	Each object in active directory exists under a certain naming context (also known as a partition). A naming context is defined as a single object in the Directory Information Tree (DIT) along with every object in the tree subordinate to it. There are three default naming contexts in Active Directory: domain, configuration, and schema.
relative_dn	oval-def:EntityStateStringType (0..1)	The relative_dn field is used to uniquely identify an object inside the specified naming context. It contains all the parts of the objects distinguished name except those outlined by the naming context.
attribute	oval-def:EntityStateStringType (0..1)	Specifies a named value contained by the object.
object_class	oval-def:EntityStateStringType (0..1)	The name of the class of which the object is an instance.
adstype	win-def:EntityStateAdstypeType (0..1)	Specifies the type of information that the specified attribute represents.
value	oval-def:EntityStateAnySimpleType (0..1)	The actual value of the specified active directory attribute.

< activedirectory57_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.11.1:1.2
- Reason: Use the original activedirectory_test. The activedirectory57_test suffers from ambiguity; it was never adequately specified, and it does not even seem possible to have structured data in the context of the enumerated AdstypeTypes. Use the original activedirectory_test instead.
- Comment: This test has been deprecated and will be removed in version 6.0 of the language.

The active directory test is used to check information about specific entries in active directory. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an activedirectory57_object and the optional state element specifies the metadata to check.

Note that this test supports complex values that are in the form of a record. For simple (string based) value collection see the activedirectory_test.

Extends: oval-def:TestType

Child Elements

Table 590: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< activedirectory57_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.11.1:1.2
- Reason: Use the original activedirectory_object. The activedirectory57_test suffers from ambiguity; it was never adequately specified, and it does not even seem possible to have structured data in the context of the enumerated AdstypeTypes. Use the original activedirectory_test instead.
- Comment: This object has been deprecated and will be removed in version 6.0 of the language.

The activedirectory57_object element is used by an active directory test to define those objects to evaluated based on a specified state. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An active directory object consists of three pieces of information, a naming context, a relative distinguished name, and an attribute. Each piece helps identify a specific active directory entry.

Note that this object supports complex values that are in the form of a record. For simple (string based) value collection see the activedirectory_object.

Extends: oval-def:ObjectType

Child Elements

Table 591: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
naming_context	win-def:EntityObjectNamingContextType (1..1)	Each object in active directory exists under a certain naming context (also known as a partition). A naming context is defined as a single object in the Directory Information Tree (DIT) along with every object in the tree subordinate to it. There are three default naming contexts in Active Directory: domain, configuration, and schema.
relative_dn	oval-def:EntityObjectStringType (1..1)	The relative_dn field is used to uniquely identify an object inside the specified naming context. It contains all the parts of the object's distinguished name except those outlined by the naming context. If the xsi:nil attribute is set to true, then the object being specified is the higher level naming context. In this case, the relative_dn element should not be collected or used in analysis. Setting xsi:nil equal to true is different than using a .* pattern match, which says to collect every relative dn under a given naming context.
attribute	oval-def:EntityObjectStringType (1..1)	Specifies a named value contained by the object. If the xsi:nil attribute is set to true, the attribute element should not be collected or used in analysis. Setting xsi:nil equal to true is different than using a .* pattern match, which says to collect every attribute under a given relative dn.
oval-def:filter	n/a (0..unbounded)	

< activedirectory57_state > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.11.1:1.2
- Reason: Use the original activedirectory_state. The activedirectory57_test suffers from ambiguity; it was never adequately specified, and it does not even seem possible to have structured data in the context of the enumerated AdstypeTypes. Use the original activedirectory_test instead.
- Comment: This state has been deprecated and will be removed in version 6.0 of the language.

The activedirectory57_state element defines the different information that can be used to evaluate the specified entries in active directory. An active directory test will reference a specific instance of this state that defines the exact settings that need to be evaluated. Please refer to the individual elements in the schema for more details about what each represents.

Note that this state supports complex values that are in the form of a record. For simple (string based) value collection see the activedirectory_state.

Extends: oval-def:StateType

Child Elements

Table 592: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
naming_context	win-def:EntityStateNamingContextType (0..1)	Each object in active directory exists under a certain naming context (also known as a partition). A naming context is defined as a single object in the Directory Information Tree (DIT) along with every object in the tree subordinate to it. There are three default naming contexts in Active Directory: domain, configuration, and schema.
relative_dn	oval-def:EntityStateStringType (0..1)	The relative_dn field is used to uniquely identify an object inside the specified naming context. It contains all the parts of the object's distinguished name except those outlined by the naming context.
attribute	oval-def:EntityStateStringType (0..1)	Specifies a named value contained by the object.
object_class	oval-def:EntityStateStringType (0..1)	The name of the class of which the object is an instance.
adstype	win-def:EntityStateAdstypeType (0..1)	The type of information that the specified attribute represents.
value	oval-def:EntityStateRecordType (0..1)	The actual value of the specified Active Directory attribute. Note that while an Active Directory attribute can contain structured data where it is necessary to collect multiple related fields that can be described by the 'record' datatype, it is not always the case. It also is possible that an Active Directory attribute can contain only a single value or an array of values. In these cases, there is not a name to uniquely identify the corresponding field which is a requirement for fields in the 'record' datatype. As a result, the name of the Active Directory attribute will be used to uniquely identify the field and satisfy this requirement.

< auditeventpolicy_test >

The auditeventpolicy_test is used to check different types of events the system should audit. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a auditeventpolicy_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 593: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< auditeventpolicy_object >

The auditeventpolicy_object element is used by an audit event policy test to define those objects to evaluate based on a specified state. There is actually only one object relating to audit event policy and this is the system as a whole. Therefore, there are no child entities defined. Any OVAL Test written to check audit event policy will reference the same auditeventpolicy_object which is basically an empty object element.

Extends: oval-def:ObjectType

< auditeventpolicy_state >

The auditeventpolicy_state element specifies the different system activities that can be audited. An audit event policy test will reference a specific instance of this state that defines the exact settings that need to be evaluated. The defined values are found in window's POLICY_AUDIT_EVENT_TYPE enumeration and accessed through the LsaQueryInformationPolicy when the InformationClass parameters are set to PolicyAuditEventsInformation. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 594: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
account_logon	win-def:EntityStateAuditType (0..1)	Audit attempts to log on to or log off of the system. Also, audit attempts to make a network connection.
account_management	win-def:EntityStateAuditType (0..1)	Audit attempts to create, delete, or change user or group accounts. Also, audit password changes.
detailed_tracking	win-def:EntityStateAuditType (0..1)	Audit specific events, such as program activation, some forms of handle duplication, indirect access to an object, and process exit. Note that this activitiy is also known as process tracking.
directory_service_access	win-def:EntityStateAuditType (0..1)	Audit attempts to access the directory service.
logon	win-def:EntityStateAuditType (0..1)	Audit attempts to log on to or log off of the system. Also, audit attempts to make a network connection.
object_access	win-def:EntityStateAuditType (0..1)	Audit attempts to access securable objects, such as files.
policy_change	win-def:EntityStateAuditType (0..1)	Audit attempts to change Policy object rules.
privilege_use	win-def:EntityStateAuditType (0..1)	Audit attempts to use privileges.
system	win-def:EntityStateAuditType (0..1)	Audit attempts to shut down or restart the computer. Also, audit events that affect system security or the security log.

< auditeventpolicysubcategories_test >

The auditeventpolicysubcategories_test is used to check the audit event policy settings on a Windows system. These settings are used to specify which system and network events are monitored. For example, if the credential_validation element has a value of AUDIT_FAILURE, it means that the system is configured to log all unsuccessful attempts to validate a user account on a system. It is important to note that these audit event policy settings are specific to certain versions of Windows. As a result, the documentation for that version of Windows should be consulted for more information on each setting. The test extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a auditeventpolicy_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 595: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< auditeventpolicysubcategories_object >

The auditeventpolicysubcategories_object element is used by an audit event policy subcategories test to define those objects to evaluate based on a specified state. There is actually only one object relating to audit event policy subcategories and this is the system as a whole. Therefore, there are no child entities defined. Any OVAL Test written to check audit event policy subcategories will reference the same auditeventpolicysubcategories_object which is basically an empty object element.

Extends: oval-def:ObjectType

< auditeventpolicysubcategories_state >

The auditeventpolicysubcategories_state element specifies the different system activities that can be audited. An audit event policy subcategories test will reference a specific instance of this state that defines the exact subcategories that need to be evaluated. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
credential_validation	win-def:EntityStateAuditType (0..1)	Audit the events produced during the validation o
kerberos_authentication_service	win-def:EntityStateAuditType (0..1)	Audit the events produced by Kerberos authentica

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
kerberos_service_ticket_operations	win-def:EntityStateAuditType (0..1)	Audit the events produced by Kerberos service tic
kerberos_ticket_events (Deprecated)	win-def:EntityStateAuditType (0..1)	Audit the events produced during the validation o
other_account_logon_events	win-def:EntityStateAuditType (0..1)	Audit the events produced by changes to user acc
application_group_management	win-def:EntityStateAuditType (0..1)	Audit the events produced by changes to applicat
computer_account_management	win-def:EntityStateAuditType (0..1)	Audit the events produced by changes to compute
distribution_group_management	win-def:EntityStateAuditType (0..1)	Audit the events produced by changes to distribut
other_account_management_events	win-def:EntityStateAuditType (0..1)	Audit the events produced by other user account c
security_group_management	win-def:EntityStateAuditType (0..1)	Audit the events produced by changes to security
user_account_management	win-def:EntityStateAuditType (0..1)	Audit the events produced by changes to user acc
dpapi_activity	win-def:EntityStateAuditType (0..1)	Audit the events produced when requests are mad
process_creation	win-def:EntityStateAuditType (0..1)	Audit the events produced when a process is crea
process_termination	win-def:EntityStateAuditType (0..1)	Audit the events produced when a process ends. T
rpc_events	win-def:EntityStateAuditType (0..1)	Audit the events produced by inbound remote pro
directory_service_access	win-def:EntityStateAuditType (0..1)	Audit the events produced when a Active Directo
directory_service_changes	win-def:EntityStateAuditType (0..1)	Audit the events produced when changes are mad
directory_service_replication	win-def:EntityStateAuditType (0..1)	Audit the events produced when two Active Direc
detailed_directory_service_replication	win-def:EntityStateAuditType (0..1)	Audit the events produced by detailed Active Dire
account_lockout	win-def:EntityStateAuditType (0..1)	Audit the events produced by a failed attempt to l
ipsec_extended_mode	win-def:EntityStateAuditType (0..1)	Audit the events produced by Internet Key Excha
ipsec_main_mode	win-def:EntityStateAuditType (0..1)	Audit the events produced by Internet Key Excha
ipsec_quick_mode	win-def:EntityStateAuditType (0..1)	Audit the events produced by Internet Key Excha
logoff	win-def:EntityStateAuditType (0..1)	Audit the events produced by closing a logon sess
logon	win-def:EntityStateAuditType (0..1)	Audit the events produced by attempts to log onto
network_policy_server	win-def:EntityStateAuditType (0..1)	Audit the events produced by RADIUS and Netw
other_logon_logoff_events	win-def:EntityStateAuditType (0..1)	Audit the events produced by other logon/logoff b
special_logon	win-def:EntityStateAuditType (0..1)	Audit the events produced by special logons. This
logon_claims	win-def:EntityStateAuditType (0..1)	Audit user and device claims information in the u
application_generated	win-def:EntityStateAuditType (0..1)	Audit the events produced by applications that us
certification_services	win-def:EntityStateAuditType (0..1)	Audit the events produced by operations on Activ
detailed_file_share	win-def:EntityStateAuditType (0..1)	Audit the events produced by attempts to access f
file_share	win-def:EntityStateAuditType (0..1)	Audit the events produced by attempts to access a
file_system	win-def:EntityStateAuditType (0..1)	Audit the events produced user attempts to access
filtering_platform_connection	win-def:EntityStateAuditType (0..1)	Audit the events produced by connections that are
filtering_platform_packet_drop	win-def:EntityStateAuditType (0..1)	Audit the events produced by packets that are dro
handle_manipulation	win-def:EntityStateAuditType (0..1)	Audit the events produced when a handle is open
kernel_object	win-def:EntityStateAuditType (0..1)	Audit the events produced by attempts to access t
other_object_access_events	win-def:EntityStateAuditType (0..1)	Audit the events produced by the management of
registry	win-def:EntityStateAuditType (0..1)	Audit the events produced by attempts to access r
sam	win-def:EntityStateAuditType (0..1)	Audit the events produced by attempts to access S
removable_storage	win-def:EntityStateAuditType (0..1)	Audit events that indicate file object access attem
central_access_policy_staging	win-def:EntityStateAuditType (0..1)	Audit events that indicate permission granted or d
audit_policy_change	win-def:EntityStateAuditType (0..1)	Audit the events produced by changes in security
authentication_policy_change	win-def:EntityStateAuditType (0..1)	Audit the events produced by changes to the auth
authorization_policy_change	win-def:EntityStateAuditType (0..1)	Audit the events produced by changes to the auth
filtering_platform_policy_change	win-def:EntityStateAuditType (0..1)	Audit the events produced by changes to the Win
mpssvc_rule_level_policy_change	win-def:EntityStateAuditType (0..1)	Audit the events produced by changes to policy ru
other_policy_change_events	win-def:EntityStateAuditType (0..1)	Audit the events produced by other security polic
non_sensitive_privilege_use	win-def:EntityStateAuditType (0..1)	Audit the events produced by the use of non-sens
other_privilege_use_events	win-def:EntityStateAuditType (0..1)	This is currently not used and has been reserved b

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
sensitive_privilege_use	win-def:EntityStateAuditType (0..1)	Audit the events produced by the use of sensitive
ipsec_driver	win-def:EntityStateAuditType (0..1)	Audit the events produced by the IPsec filter driv
other_system_events	win-def:EntityStateAuditType (0..1)	Audit the events produced by the startup and shut
security_state_change	win-def:EntityStateAuditType (0..1)	Audit the events produced by changes in the secu
security_system_extension	win-def:EntityStateAuditType (0..1)	Audit the events produced by the security system
system_integrity	win-def:EntityStateAuditType (0..1)	Audit the events that indicate that the integrity se
group_membership	win-def:EntityStateAuditType (0..1)	This subcategory audits the group membership of
pnp_activity	win-def:EntityStateAuditType (0..1)	This subcategory audits events generated by plug
user_device_claims	win-def:EntityStateAuditType (0..1)	This subcategory audits the user and device claim
audit_detailedtracking_tokenrightadjusted	win-def:EntityStateAuditType (0..1)	This subcategory audits when token privileges ar

< cmdlet_test >

The cmdlet_test is used to levarage a PowerShell cmdlet to check a Windows system. The test extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a cmdlet_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 597: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< cmdlet_object >

The cmdlet_object element is used by a cmdlet_test to identify the set of cmdlets to use and the parameters to provide to them for checking the state of a system. In order to ensure the consistency of PowerShell cmdlet support among OVAL interpreters as well as ensure that the state of a system is not changed, every OVAL interpreter must implement the following requirements. An OVAL interpreter must only support the processing of the verbs specified in the EntityObjectCmdletVerbType. If a cmdlet verb that is not defined in this enumeration is discovered, an error should be reported and the cmdlet must not be executed on the system. While XML Schema validation will enforce this requirement, it is strongly recommended that OVAL interpreters implement a whitelist of allowed cmdlets. This can be done using constrained runspaces which can limit the PowerShell execution environment. For more information, please see Microsoft's documentation on Windows PowerShell Host Application Concepts. Furthermore, it is strongly recommended that OVAL interpreters also implement PowerShell support with the NoLanguage mode enabled. The NoLanguage mode ensures that scripts that need to be evaluated are not allowed in the runspace. For more information about the NoLanguage mode, please see Microsoft's documentation on the PSLanguageMode enumeration.

Extends: oval-def:ObjectType

Child Elements

Table 598: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
module_name	oval-def:EntityObjectStringType (1..1)	The name of the module that contains the cmdlet.
module_id	win-def:EntityObjectGUIDType (1..1)	The globally unique identifier for the module. If xsi:nil=’true’, it does not matter which module the command comes from.
module_version	oval-def:EntityObjectVersionType (1..1)	The version of the module that contains the cmdlet in the form of MAJOR.MINOR. If xsi:nil=’true’, that implies it does not matter which version of the module the command refers to.
verb	win-def:EntityObjectCmdletVerbType (1..1)	The cmdlet verb.
noun	oval-def:EntityObjectStringType (1..1)	The cmdlet noun.
parameters	oval-def:EntityObjectRecordType (1..1)	A list of properties (name and value pairs) as input to invoke the cmdlet. Each property name must be unique. When xsi:nil=’true’, parameters are not provided to the cmdlet.
select	oval-def:EntityObjectRecordType (1..1)	A list of fields (name and value pairs) used as input to the Select-Object cmdlet to select specific output properties. Each property name must be unique. Please note that the use of the ‘*’ character, to select all properties, is not permitted. This is because the value record entity, in the state and item, require unique field name values to ensure that any query results can be evaluated consistently. This is equivalent to piping the output of a cmdlet to the Select-Object cmdlet. When xsi:nil=’true’, the Select-Object is not used.
oval-def:filter	n/a (0..unbounded)	

< cmdlet_state >

The cmdlet_state allows for assertions about the presence of PowerShell cmdlet related properties and values obtained from a cmdlet.

Extends: oval-def:StateType

Child Elements

Table 599: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
module_name	oval-def:EntityStateStringType (0..1)	The name of the module that contains the cmdlet.
module_id	win-def:EntityStateGUIDType (0..1)	The globally unique identifier for the module.
module_version	oval-def:EntityStateVersionType (0..1)	The version of the module that contains the cmdlet in the form of MAJOR.MINOR.
verb	win-def:EntityStateCmdletVerbType (0..1)	The cmdlet verb.
noun	oval-def:EntityStateStringType (0..1)	The cmdlet noun.
parameters	oval-def:EntityStateRecordType (0..1)	A list of properties (name and value pairs) as input to invoke the cmdlet. Each property name must be unique.
select	oval-def:EntityStateRecordType (0..1)	A list of fields (name and value pairs) used as input to the Select-Object cmdlet to select specific output properties. Each property name must be unique.
value	oval-def:EntityStateRecordType (0..1)	The expected value represented as a set of fields (name and value pairs). Each field must be have a unique name.

< dnscache_test >

The dnscache_test is used to check the time to live and IP addresses associated with a domain name. The time to live and IP addresses for a particular domain name are retrieved from the DNS cache on the local system. The entries in the DNS cache can be collected using Microsoft's DnsGetCacheDataTable() and DnsQuery() API calls. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a dnscache_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 600: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< dnscache_object >

The dnscache_object is used by the dnscache_test to specify the domain name(s) that should be collected from the DNS cache on the local system. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 601: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
domain_name	oval-def:EntityObjectStringType (1..1)	The domain_name element specifies the domain name(s) that should be collected from the DNS cache on the local system.
oval-def:filter	n/a (0..unbounded)	

< dnscache_state >

The dnscache_state contains three entities that are used to check the domain name, time to live, and IP addresses associated with the DNS cache entry.

Extends: oval-def:StateType

Child Elements

Table 602: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
domain_name	oval-def:EntityStateStringType (0..1)	The domain_name element contains a string that represents a domain name that was collected from the DNS cache on the local system.
ttl	oval-def:EntityStateIntType (0..1)	The ttl element contains an integer that represents the time to live in seconds of the DNS cache entry.
ip_address	oval-def:EntityStateIPAddressStringType (0..1)	The ip_address element contains a string that represents an IP address associated with the specified domain name that was collected from the DNS cache on the local system. Note that the IP address can be IPv4 or IPv6.

< file_test >

The file test is used to check metadata associated with Windows files. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a file_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 603: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< file_object >

The file_object element is used by a file test to define the specific file(s) to be evaluated. The file_object will collect directories and all Windows file types (FILE_TYPE_CHAR, FILE_TYPE_DISK, FILE_TYPE_PIPE, FILE_TYPE_REMOTE, and FILE_TYPE_UNKNOWN). Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A file object defines the path and filename or complete filepath of the file(s). In addition, a number of behaviors may be provided that help guide the collection of objects. Please refer to the FileBehaviors complex type for more information about specific behaviors.

The set of files to be evaluated may be identified with either a complete filepath or a path and filename. Only one of these options may be selected.

It is important to note that 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. This is because the ‘filepath’ element represents an absolute path to a particular file and it is not possible to recurse over a file.

Extends: oval-def:ObjectType

Child Elements

Table 604: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	win-def:FileBehaviors (0..1)	
filepath	oval-def:EntityObjectStringType (1..1)	The filepath element specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath.
path	oval-def:EntityObjectStringType (1..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-def:EntityObjectStringType (1..1)	The filename element specifies the name of a file to evaluate. If the xsi:nil attribute is set to true, then the object being specified is the higher level directory object (not all the files in the directory). In this case, the filename element should not be used during collection and would result in the unique set of items being the directories themselves. For example, one would set xsi:nil to true if the desire was to test the attributes or permissions associated with a directory. Setting xsi:nil equal to true is different than using a .* pattern match, which says to collect every file under a given path.
oval-def:filter	n/a (0..unbounded)	

< file_state >

The file_state element defines the different metadata associate with a Windows file. This includes the path, filename, owner, size, last modified time, version, etc. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 605: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-def:EntityStateStringType (0..1)	The filepath element specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath.
path	oval-def:EntityStateStringType (0..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-def:EntityStateStringType (0..1)	The filename element specifies the name of the file.
owner	oval-def:EntityStateStringType (0..1)	The owner element is a string that contains the name of the owner. The name should be specified in the DOMAINusername format.
size	oval-def:EntityStateIntType (0..1)	The size element is the size of the file in bytes.
a_time	oval-def:EntityStateIntType (0..1)	Time of last access of file. Valid on NTFS but not on FAT formatted disk drives. The string should represent the FILETIME structure which is a 64-bit value representing the number of 100-nanosecond intervals since January 1, 1601 (UTC).
c_time	oval-def:EntityStateIntType (0..1)	Time of creation of file. Valid on NTFS but not on FAT formatted disk drives. The string should represent the FILETIME structure which is a 64-bit value representing the number of 100-nanosecond intervals since January 1, 1601 (UTC).
m_time	oval-def:EntityStateIntType (0..1)	Time of last modification of file. The string should represent the FILETIME structure which is a 64-bit value representing the number of 100-nanosecond intervals since January 1, 1601 (UTC).
ms_checksum	oval-def:EntityStateStringType (0..1)	The checksum of the file as supplied by Microsoft's MapFileAndCheckSum function.
version	oval-def:EntityStateVersionType (0..1)	The version element is the delimited version string of the file.
type	win-def:EntityStateFileTypeType (0..1)	The type element marks whether the file is a named pipe, standard file, etc. These types are the return values for GetFileType. For directories, this element must have a status of 'does not exist'.
attribute	win-def:EntityStateFileAttributeType (0..1)	The attribute element marks a Windows file attribute. These types are the return values for GetFileAttribute.The attribute element can be included multiple times in a system characteristic item in order to record that a file has a number of different attributes. Note that the entity_check attribute associated with EntityStateStringType guides the evaluation of entities like the attribute entity that refer to items that can occur an unbounded number of times.
development_class	oval-def:EntityStateStringType (0..1)	The development_class element allows the distinction to be made between the GDR development environment and the QFE development environment. This field holds the text found in front of the mmmmmm-nnnn version, for example srv03_gdr.
company	oval-def:EntityStateStringType (0..1)	This entity defines a company name to be found within the version-information structure.
internal_name	oval-def:EntityStateStringType (0..1)	This entity defines an internal name to be found within the version-information structure.
language	oval-def:EntityStateStringType (0..1)	This entity defines a language to be found within the version-information structure.

== FileBehaviors ==

The FileBehaviors complex type defines a number of behaviors that allow a more detailed definition of the file_object being specified. 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.

It is important to note that 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. This is because the 'filepath' element represents an absolute path to a particular file and it is not possible to recurse over a file.

Attributes

Table 606: Attributes

Attribute	Type	Desc.
max_depth	Restriction of xsd:integer (optional *default*='-1')	'max_depth' defines the maximum depth of recursion to perform when a recurse_direction is specified. A value of '0' is equivalent to no recursion, '1' means to step only one directory level up/down, and so on. The default value is '-1' meaning no limitation. For a 'max_depth' of -1 or any value of 1 or more the starting directory must be considered in the recursive search.

Note that the default recurse_direction behavior is 'none' so even though max_depth specifies no limitation by default, the recurse_direction behavior turns recursion off. Note that this behavior only applies with the equality operation on the path entity.

- - recurse
  - Restriction of xsd:string (optional *default*='directories') ('directories', 'junctions', 'junctions and directories')
  - 'recurse' defines how to recurse into the path entity, in other words what to follow during recursion. Options include junctions, directories, or both (a junction on Windows is equivalent to a symlink on Unix). Note that a max-depth other than 0 has to be specified for recursion to take place and for this attribute to mean anything.

Note that this behavior only applies with the equality operation on the path entity.

- - recurse_direction
  - Restriction of xsd:string (optional *default*='none') ('none', 'up', 'down')
  - 'recurse_direction' defines the direction, either 'up' to parent directories, or 'down' into child directories to recursively search for files. When recursing up or down, one is limited by the max_depth behavior. Note that it is not an error if max_depth specifies a certain level of recursion and that level does not exist. Recursing should only go as deep as available. The default value is 'none' for no recursion.

Note that this behavior only applies with the equality operation on the path entity.

- - recurse_file_system
  - Restriction of xsd:string (optional *default*='all') ('all', 'local', 'defined')
  - 'recurse_file_system' defines the file system limitation of any searching and applies to all operations as specified on the path or filepath entity. The value of 'local' limits the search scope to local file

systems (as opposed to file systems mounted from an external system). The value of 'defined' keeps any recursion within the file system that the file_object (path+filename or filepath) has specified. For example, if the path specified was "C:", you would search only the C: drive, not other filesystems mounted to descendant paths. The value of 'defined' only applies when an equality operation is used for searching because the path or filepath entity must explicitly define a file system. The default value is 'all' meaning to search all available file systems for data collection.

Note that in most cases it is recommended that the value of 'local' be used to ensure that file system searching is limited to only t

-   - windows_view
    - Restriction of xsd:string (optional *default*='64_bit') ('32_bit', '64_bit')
    - 64-bit versions of Windows provide an alternate file system and registry views to 32-bit applications. This behavior allows the OVAL Object to state which view should be examined. This behavior only applies to 64-bit Windows, and must not be applied on other platforms.

Note that the values have the following meaning: '64_bit' - Indicates that the 64-bit view on 64-bit Windows operating systems must be examined. On a 32-bit system, the Object must be evaluated without applying the behavior. '32_bit' - Indicates that the 32-bit view must be examined. On a 32-bit system, the Object must be evaluated without applying the behavior. It is recommended that the corresponding 'windows_view' entity be set on the OVAL Items that are collected when this behavior is used to distinguish between OVAL Items that were collected in the 32-bit or 64-bit views.

< fileauditedpermissions53_test >

The file audit permissions test is used to check the audit permissions associated with Windows files. Note that the trustee's audited permissions are the audit permissons that the SACL grants to the trustee or to any groups of which the trustee is a member. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a fileauditedpermissions_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 607: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< fileauditedpermissions53_object >

The fileauditedpermissions53_object element is used by a file audited permissions test to define the objects used to evalutate against the specified state. The fileauditedpermissions53_object will collect directories and all Windows file types (FILE_TYPE_CHAR, FILE_TYPE_DISK, FILE_TYPE_PIPE, FILE_TYPE_REMOTE, and FILE_TYPE_UNKNOWN). Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic.

A fileauditedpermissions53_object is defined as a combination of a Windows file and trustee SID. The file represents the file to be evaluated while the trustee SID represents the account (SID) to check audited permissions of. If multiple files or SIDs are matched by either reference, then each possible combination of file and SID is a matching file audited permissions object. In addition, a number of behaviors may be provided that help guide the collection of objects. Please refer to the FileAuditPermissions53Behaviors complex type for more information about specific behaviors.

The set of files to be evaluated may be identified with either a complete filepath or a path and filename. Only one of these options may be selected.

It is important to note that 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. This is because the 'filepath' element represents an absolute path to a particular file and it is not possible to recurse over a file.

Extends: oval-def:ObjectType

Child Elements

Table 608: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	win-def:FileAuditPermissions53Behaviors (0..1)	
filepath	oval-def:EntityObjectStringType (1..1)	The filepath element specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath.
path	oval-def:EntityObjectStringType (1..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-def:EntityObjectStringType (1..1)	The filename element specifies the name of a file to evaluate. If the xsi:nil attribute is set to true, then the object being specified is the higher level directory object (not all the files in the directory). In this case, the filename element should not be used during collection and would result in the unique set of items being the directories themselves. For example, one would set xsi:nil to true if the desire was to test the attributes or permissions associated with a directory. Setting xsi:nil equal to true is different than using a .* pattern match, which says to collect every file under a given path.
trustee_sid	oval-def:EntityObjectStringType (1..1)	The trustee_sid entity identifies a unique SID associated with a user, group, system, or program (such as a Windows service). If an operation other than equals is used to identify matching trustees (i.e. not equal, or a pattern match) then the resulting matches shall be limited to only the trustees referenced in the file's Security Descriptor. The scope is limited here to avoid unnecessarily resource intensive searches for trustees. Note that the larger scope of all known trustees may be obtained through the use of variables.
oval-def:filter	n/a (0..unbounded)	

< fileauditedpermissions53_state >

The fileauditedpermissions53_state element defines the different audit permissions that can be associated with a given fileauditedpermissions53_object. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 609: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-def:EntityStateStringType (0..1)	The filepath element specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath.
path	oval-def:EntityStateStringType (0..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-def:EntityStateStringType (0..1)	The filename element specifies the name of a file to test for.
trustee_sid	oval-def:EntityStateStringType (0..1)	The trustee_sid element is the unique SID that associated a user, group, system, or program (such as a Windows service).
standard_delete	win-def:EntityStateAuditType (0..1)	The right to delete the object.
standard_read_control	win-def:EntityStateAuditType (0..1)	The right to read the information in the object's Security Descriptor, not including the information in the SACL.
standard_write_dac	win-def:EntityStateAuditType (0..1)	The right to modify the DACL in the object's Security Descriptor.
standard_write_owner	win-def:EntityStateAuditType (0..1)	The right to change the owner in the object's Security Descriptor.
standard_synchronize	win-def:EntityStateAuditType (0..1)	The right to use the object for synchronization. This enables a thread to wait until the object is in the signaled state. Some object types do not support this access right.
access_system_security	win-def:EntityStateAuditType (0..1)	Indicates access to a system access control list (SACL).
generic_read	win-def:EntityStateAuditType (0..1)	Read access.
generic_write	win-def:EntityStateAuditType (0..1)	Write access.
generic_execute	win-def:EntityStateAuditType (0..1)	Execute access.
generic_all	win-def:EntityStateAuditType (0..1)	Read, write, and execute access.
file_read_data	win-def:EntityStateAuditType (0..1)	Grants the right to read data from the file.
file_write_data	win-def:EntityStateAuditType (0..1)	Grants the right to write data to the file.
file_append_data	win-def:EntityStateAuditType (0..1)	Grants the right to append data to the file.
file_read_ea	win-def:EntityStateAuditType (0..1)	Grants the right to read extended attributes.

== FileAuditPermissions53Behaviors ==

The FileAuditPermissions53Behaviors complex type defines a number of behaviors that allow a more detailed definition of the fileauditpermissions53_object being specified. 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.

It is important to note that 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. This is because the 'filepath' element represents an absolute path to a particular file and it is not possible to recurse over a file.

The FileAuditPermissions53Behaviors extend the win-def:FileBehaviors and therefore include the behaviors defined by that type.

Extends: win-def:FileBehaviors

Attributes

Table 610: Attributes

Attribute	Type	Desc.
include_group (Deprecated)	xsd:boolean (optional *default*='true')	'include_group' defines whether the group SID should be included in the object when the object is defined by a group SID. For example, the intent of an object defined by a group SID might be to retrieve all the user SIDs that are a member of the group, but not the group SID itself.
resolve_group (Deprecated)	xsd:boolean (optional *default*='false')	The 'resolve_group' behavior defines whether an object set defined by a group SID should be resolved to return a set that contains all the user SIDs that are a member of that group. Note that all child groups should also be resolved any valid domain users that are members of the group should also be included. The intent of this behavior is to end up with a list of all individual users from that system that make up the group once everything has been resolved.

< fileauditedpermissions_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.3
- Reason: Replaced by the fileauditedpermissions53_test. This test uses a trustee_name element for identifying trustees. Trustee names are not unique, and a new test was created to use trustee SIDs, which are unique. See the fileauditedpermissions53_test.
- Comment: This test has been deprecated and will be removed in version 6.0 of the language.

The file audited permissions test is used to check the audit permissions associated with Windows files. Note that the trustee's audited permissions are the audit permissons that the SACL grants to the trustee or to any groups of which the trustee is a member. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a fileauditedpermissions_object, and the optional state element references a fileauditedpermissions_state that specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 611: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< fileauditedpermissions_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.3
- Reason: Replaced by the fileauditedpermissions53_object. This object uses a trustee_name element for identifying trustees. Trustee names are not unique, and a new object was created to use trustee SIDs, which are unique. See the fileauditedpermissions53_object.
- Comment: This object has been deprecated and will be removed in version 6.0 of the language.

The fileauditedpermissions_object element is used by a file audited permissions test to define the objects used to evalutate against the specified state. The fileauditedpermissions_object will collect directories and all Windows file types (FILE_TYPE_CHAR, FILE_TYPE_DISK, FILE_TYPE_PIPE, FILE_TYPE_REMOTE, and FILE_TYPE_UNKNOWN). Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic.

A fileauditedpermissions_object is defined as a combination of a Windows file and trustee name. The file represents the file to be evaluated while the trustee name represents the account (SID) to check audited permissions of. If multiple files or SIDs are matched by either reference, then each possible combination of file and SID is a matching file audited permissions object. In addition, a number of behaviors may be provided that help guide the collection of objects. Please refer to the FileAuditPermissionsBehaviors complex type for more information about specific behaviors.

Extends: oval-def:ObjectType

Child Elements

Table 612: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	win-def:FileAuditPermissionsBehaviors (0..1)	
path	oval-def:EntityObjectStringType (1..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-def:EntityObjectStringType (1..1)	The filename element specifies the name of a file to evaluate. If the xsi:nil attribute is set to true, then the object being specified is the higher level directory object (not all the files in the directory). In this case, the filename element should not be used during collection and would result in the unique set of items being the directories themselves. For example, one would set xsi:nil to true if the desire was to test the attributes or permissions associated with a directory. Setting xsi:nil equal to true is different than using a .* pattern match, which says to collect every file under a given path.
trustee_name	oval-def:EntityObjectStringType (1..1)	The trustee_name element is the unique name that associated a particular SID. A SID can be associated with a user, group, or program (such as a Windows service). In Windows, trustee names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, trustee names should be identified in the form: "domaintrustee name". For local trustee names use: "computer nametrustee name". For built-in accounts on the system, use the trustee name without a domain.

< fileauditedpermissions_state > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.3
- Reason: Replaced by the fileauditedpermissions53_state. This state uses a trustee_name element for identifying trustees. Trustee names are not unique, and a new state was created to use trustee SIDs, which are unique. See the fileauditedpermissions53_state.
- Comment: This state has been deprecated and will be removed in version 6.0 of the language.

The fileauditedpermissions_state element defines the different audit permissions that can be associated with a given fileauditedpermissions_object. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 613: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
path	oval-def:EntityStateStringType (0..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-def:EntityStateStringType (0..1)	The filename element specifies the name of a file to test for.
trustee_name	oval-def:EntityStateStringType (0..1)	The trustee_name is the unique name associated with a particular security identifier (SID). In Windows, trustee names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, trustee names should be identified in the form: "domaintrustee name". For local trustee names use: "computer nametrustee name". For built-in accounts on the system, use the trustee name without a domain.
standard_delete	win-def:EntityStateAuditType (0..1)	The right to delete the object.
standard_read_control	win-def:EntityStateAuditType (0..1)	The right to read the information in the object's Security Descriptor, not including the information in the SACL.
standard_write_dac	win-def:EntityStateAuditType (0..1)	The right to modify the DACL in the object's Security Descriptor.
standard_write_owner	win-def:EntityStateAuditType (0..1)	The right to change the owner in the object's Security Descriptor.
standard_synchronize	win-def:EntityStateAuditType (0..1)	The right to use the object for synchronization. This enables a thread to wait until the object is in the signaled state. Some object types do not support this access right.
access_system_security	win-def:EntityStateAuditType (0..1)	Indicates access to a system access control list (SACL).
generic_read	win-def:EntityStateAuditType (0..1)	Read access.
generic_write	win-def:EntityStateAuditType (0..1)	Write access.
generic_execute	win-def:EntityStateAuditType (0..1)	Execute access.
generic_all	win-def:EntityStateAuditType (0..1)	Read, write, and execute access.
file_read_data	win-def:EntityStateAuditType (0..1)	Grants the right to read data from the file.
file_write_data	win-def:EntityStateAuditType (0..1)	Grants the right to write data to the file.
file_append_data	win-def:EntityStateAuditType (0..1)	Grants the right to append data to the file.
file_read_ea	win-def:EntityStateAuditT	Grants the right to read extended attributes.

== FileAuditPermissionsBehaviors == (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.3
- Reason: Replaced by the FileAuditPermissionsBehaviors53. The FileAuditPermissionsBehaviors complex type is used by the fileauditedpermissions_test which uses a trustee_name element for identifying trustees. Trustee names are not unique, and a new test was created to use trustee SIDs, which are unique. This new test utilizes the FileAuditPermissionsBehaviors53 complex type, and as a result, the FileAuditPermissionsBehaviors complex type is no longer needed.
- Comment: This complex type has been deprecated and will be removed in version 6.0 of the language.

The FileAuditPermissionsBehaviors complex type defines a number of behaviors that allow a more detailed definition of the fileauditpermissions_object being specified. 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.

The FileAuditPermissionsBehaviors extend the win-def:FileBehaviors and therefore include the behaviors defined by that type.

Extends: win-def:FileBehaviors

Attributes

Table 614: Attributes

Attribute	Type	Desc.
include_group (Deprecated)	xsd:boolean (optional *default*='true')	'include_group' defines whether the group trustee name should be included in the object when the object is defined by a group trustee name. For example, the intent of an object defined by a group trustee name might be to retrieve all the user SIDs that are a member of the group, but not the group trustee name itself.
resolve_group (Deprecated)	xsd:boolean (optional *default*='false')	The 'resolve_group' behavior defines whether an object set defined by a group SID should be resolved to return a set that contains all the user SIDs that are a member of that group. Note that all child groups should also be resolved any valid domain users that are members of the group should also be included. The intent of this behavior is to end up with a list of all individual users from that system that make up the group once everything has been resolved.

< fileeffectiverights53_test >

The file effective rights test is used to check the effective rights associated with Windows files. Note that the trustee's effective access rights are the access rights that the DACL grants to the trustee or to any groups of which the trustee is a member. The fileeffectiverights53_test element extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a fileeffectiverights53_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 615: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< fileeffectiverights53_object >

The fileeffectiverights53_object element is used by a file effective rights test to define the objects used to evalutate against the specified state. The fileeffectiverights53_object will collect directories and all Windows file types (FILE_TYPE_CHAR, FILE_TYPE_DISK, FILE_TYPE_PIPE, FILE_TYPE_REMOTE, and FILE_TYPE_UNKNOWN). Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic.

A fileeffectiverights53_object is defined as a combination of a Windows file and trustee SID. The file represents the file to be evaluated while the trustee SID represents the account (SID) to check effective rights of. If multiple files or SIDs are matched by either reference, then each possible combination of file and SID is a matching file effective rights object. In addition, a number of behaviors may be provided that help guide the collection of objects. Please refer to the FileEffectiveRights53Behaviors complex type for more information about specific behaviors.

The set of files to be evaluated may be identified with either a complete filepath or a path and filename. Only one of these options may be selected.

It is important to note that 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. This is because the ‘filepath’ element represents an absolute path to a particular file and it is not possible to recurse over a file.

Extends: oval-def:ObjectType

Child Elements

Table 616: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	win-def:FileEffectiveRights53Behaviors (0..1)	
filepath	oval-def:EntityObjectStringType (1..1)	The filepath element specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath.
path	oval-def:EntityObjectStringType (1..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-def:EntityObjectStringType (1..1)	The filename element specifies the name of a file to evaluate. If the xsi:nil attribute is set to true, then the object being specified is the higher level directory object (not all the files in the directory). In this case, the filename element should not be used during collection and would result in the unique set of items being the directories themselves. For example, one would set xsi:nil to true if the desire was to test the attributes or permissions associated with a directory. Setting xsi:nil equal to true is different than using a .* pattern match, which says to collect every file under a given path..
trustee_sid	oval-def:EntityObjectStringType (1..1)	The trustee_sid entity identifies a unique SID associated with a user, group, system, or program (such as a Windows service). If an operation other than equals is used to identify matching trustees (i.e. not equal, or a pattern match) then the resulting matches shall be limited to only the trustees referenced in the file's Security Descriptor. The scope is limited here to avoid unnecessarily resource intensive searches for trustees. Note that the larger scope of all known trustees may be obtained through the use of variables.
oval-def:filter	n/a (0..unbounded)	

< fileeffectiverights53_state >

The fileeffectiverights53_state element defines the different rights that can be associated with a given fileeffectiverights53_object. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 617: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-def:EntityStateStringType (0..1)	The filepath element specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath.
path	oval-def:EntityStateStringType (0..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-def:EntityStateStringType (0..1)	The filename element specifies the name of the file.
trustee_sid	oval-def:EntityStateStringType (0..1)	The trustee_sid element is the unique SID that associated a user, group, system, or program (such as a Windows service).
standard_delete	oval-def:EntityStateBoolType (0..1)	The right to delete the object.
standard_read_control	oval-def:EntityStateBoolType (0..1)	The right to read the information in the object's Security Descriptor, not including the information in the SACL.
standard_write_dac	oval-def:EntityStateBoolType (0..1)	The right to modify the DACL in the object's Security Descriptor.
standard_write_owner	oval-def:EntityStateBoolType (0..1)	The right to change the owner in the object's Security Descriptor.
standard_synchronize	oval-def:EntityStateBoolType (0..1)	The right to use the object for synchronization. This enables a thread to wait until the object is in the signaled state. Some object types do not support this access right.
access_system_security	oval-def:EntityStateBoolType (0..1)	Indicates access to a system access control list (SACL).
generic_read	oval-def:EntityStateBoolType (0..1)	Read access.
generic_write	oval-def:EntityStateBoolType (0..1)	Write access.
generic_execute	oval-def:EntityStateBoolType (0..1)	Execute access.
generic_all	oval-def:EntityStateBoolType (0..1)	Read, write, and execute access.
file_read_data	oval-def:EntityStateBoolType (0..1)	Grants the right to read data from the file, or if a directory, grants the right to list the contents of the directory.
file_write_data	oval-def:EntityStateBoolType (0..1)	Grants the right to write data to the file, or if a directory, grants the right to add a file to the directory.
file_append_data	oval-def:EntityStateBoolType (0..1)	Grants the right to append data to the file, or if a directory, grants the right to add a sub-directory to the directory.
file_read_ea	oval-def:EntityStateBoolType (0..1)	Grants the right to read extended attributes.

== FileEffectiveRights53Behaviors ==

The FileEffectiveRights53Behaviors complex type defines a number of behaviors that allow a more detailed definition of the fileeffectiverights53_object being specified. 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.

It is important to note that 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. This is because the 'filepath' element represents an absolute path to a particular file and it is not possible to recurse over a file.

The FileEffectiveRights53Behaviors extend the win-def:FileBehaviors and therefore include the behaviors defined by that type.

Extends: win-def:FileBehaviors

Attributes

Table 618: Attributes

Attribute	Type	Desc.
include_group (Deprecated)	xsd:boolean (optional *default*='true')	'include_group' defines whether the group SID should be included in the object when the object is defined by a group SID. For example, the intent of an object defined by a group SID might be to retrieve all the user SIDs that are a member of the group, but not the group SID itself.
resolve_group (Deprecated)	xsd:boolean (optional *default*='false')	The 'resolve_group' behavior defines whether an object set defined by a group SID should be resolved to return a set that contains all the user SIDs that are a member of that group. Note that all child groups should also be resolved any valid domain users that are members of the group should also be included. The intent of this behavior is to end up with a list of all individual users from that system that make up the group once everything has been resolved.

< fileeffectiverights_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.3
- Reason: Replaced by the fileeffectiverights53_test. This test uses a trustee_name element for identifying trustees. Trustee names are not unique, and a new test was created to use trustee SIDs, which are unique. See the fileeffectiverights53_test.
- Comment: This test has been deprecated and will be removed in version 6.0 of the language.

The file effective rights test is used to check the effective rights associated with Windows files. Note that the trustee's effective access rights are the access rights that the DACL grants to the trustee or to any groups of which the trustee is a member. The fileeffectiverights_test element extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a fileeffectiverights_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 619: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< fileeffectiverights_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.3
- Reason: Replaced by the fileeffectiverights_object. This object uses a trustee_name element for identifying trustees. Trustee names are not unique, and a new object was created to use trustee SIDs, which are unique. See the fileeffectiverights53_object.
- Comment: This object has been deprecated and will be removed in version 6.0 of the language.

The fileeffectiverights_object element is used by a file effective rights test to define the objects used to evalutate against the specified state. The fileeffectiverights_object will collect directories and all Windows file types (FILE_TYPE_CHAR, FILE_TYPE_DISK, FILE_TYPE_PIPE, FILE_TYPE_REMOTE, and FILE_TYPE_UNKNOWN). Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic.

A fileeffectiverights_object is defined as a combination of a Windows file and trustee name. The file represents the file to be evaluated while the trustee name represents the account (SID) to check effective rights of. If multiple files or SIDs are matched by either reference, then each possible combination of file and SID is a matching file effective rights object. In addition, a number of behaviors may be provided that help guide the collection of objects. Please refer to the FileEffectiveRightsBehaviors complex type for more information about specific behaviors.

Extends: oval-def:ObjectType

Child Elements

Table 620: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	win-def:FileEffectiveRightsBehaviors (0..1)	
path	oval-def:EntityObjectStringType (1..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-def:EntityObjectStringType (1..1)	The filename element specifies the name of a file to evaluate. If the xsi:nil attribute is set to true, then the object being specified is the higher level directory object (not all the files in the directory). In this case, the filename element should not be used during collection and would result in the unique set of items being the directories themselves. For example, one would set xsi:nil to true if the desire was to test the attributes or permissions associated with a directory. Setting xsi:nil equal to true is different than using a .* pattern match, which says to collect every file under a given path.
trustee_name	oval-def:EntityObjectStringType (1..1)	The trustee_name element is the unique name that associated a particular SID. A SID can be associated with a user, group, or program (such as a Windows service). In Windows, trustee names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, trustee names should be identified in the form: “domaintrustee name”. For local trustee names use: “computer nametrustee name”. For built-in accounts on the system, use the trustee name without a domain.

< fileeffectiverights_state > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.3
- Reason: Replaced by the fileeffectiverights53_state. This state uses a trustee_name element for identifying trustees. Trustee names are not unique, and a new state was created to use trustee SIDs, which are unique. See the fileeffectiverights53_state.
- Comment: This state has been deprecated and will be removed in version 6.0 of the language.

The fileeffectiverights_state element defines the different rights that can be associated with a given fileeffectiverights_object. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 621: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
path	oval-def:EntityStateStringType (0..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-def:EntityStateStringType (0..1)	The filename element specifies the name of the file.
trustee_name	oval-def:EntityStateStringType (0..1)	The unique name associated with a particular security identifier (SID). In Windows, trustee names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, trustee names should be identified in the form: “domaintrustee name”. For local trustee names use: “computer nametrustee name”. For built-in accounts on the system, use the trustee name without a domain.
standard_delete	oval-def:EntityStateBoolType (0..1)	The right to delete the object.
standard_read_control	oval-def:EntityStateBoolType (0..1)	The right to read the information in the object’s Security Descriptor, not including the information in the SACL.
standard_write_dac	oval-def:EntityStateBoolType (0..1)	The right to modify the DACL in the object’s Security Descriptor.
standard_write_owner	oval-def:EntityStateBoolType (0..1)	The right to change the owner in the object’s Security Descriptor.
standard_synchronize	oval-def:EntityStateBoolType (0..1)	The right to use the object for synchronization. This enables a thread to wait until the object is in the signaled state. Some object types do not support this access right.
access_system_security	oval-def:EntityStateBoolType (0..1)	Indicates access to a system access control list (SACL).
generic_read	oval-def:EntityStateBoolType (0..1)	Read access.
generic_write	oval-def:EntityStateBoolType (0..1)	Write access.
generic_execute	oval-def:EntityStateBoolType (0..1)	Execute access.
generic_all	oval-def:EntityStateBoolType (0..1)	Read, write, and execute access.
file_read_data	oval-def:EntityStateBoolType (0..1)	Grants the right to read data from the file, or if a directory, grants the right to list the contents of the directory.
file_write_data	oval-def:EntityStateBoolType (0..1)	Grants the right to write data to the file, or if a directory, grants the right to add a file to the directory.
file_append_data	oval-def:EntityStateBoolType (0..1)	Grants the right to append data to the file, or if a directory, grants the right to add a subdirectory to the directory.
file_read_ea	oval-def:EntityStateBoolType (0..1)	Grants the right to read extended attributes.

== FileEffectiveRightsBehaviors == (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.3
- Reason: Replaced by the FileEffectiveRightsBehaviors53. The FileEffectiveRightsBehaviors complex type is used by the fileeffectiverights_test which uses a trustee_name element for identifying trustees. Trustee names are not unique, and a new test was created to use trustee SIDs, which are unique. This new test utilizes the FileEffectiveRightsBehaviors53 complex type, and as a result, the FileEffectiveRightsBehaviors complex type is no longer needed.
- Comment: This complex type has been deprecated and will be removed in version 6.0 of the language.

The FileEffectiveRightsBehaviors complex type defines a number of behaviors that allow a more detailed definition of the fileeffectiverights_object being specified. 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.

The FileEffectiveRightsBehaviors extend the win-def:FileBehaviors and therefore include the behaviors defined by that type.

Extends: win-def:FileBehaviors

Attributes

Table 622: Attributes

Attribute	Type	Desc.
include_group (Deprecated)	xsd:boolean (optional *default*='true')	'include_group' defines whether the group trustee name should be included in the object when the object is defined by a group trustee name. For example, the intent of an object defined by a group SID might be to retrieve all the user trustee names that are members of the group, but not the group trustee name itself.
resolve_group (Deprecated)	xsd:boolean (optional *default*='false')	The 'resolve_group' behavior defines whether an object set defined by a group SID should be resolved to return a set that contains all the user SIDs that are a member of that group. Note that all child groups should also be resolved any valid domain users that are members of the group should also be included. The intent of this behavior is to end up with a list of all individual users from that system that make up the group once everything has been resolved.

< group_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.11
- Reason: Replaced by the group_sid_test. This test uses trustee names for identifying accounts on the system. Trustee names are not unique and the group_sid_test, which uses trustee SIDs which are unique, should be used instead. See the group_sid_test.
- Comment: This test has been deprecated and will be removed in version 6.0 of the language.

The group_test allows the different users and subgroups, that directly belong to specific groups (identified by name), to be tested. When the group_test collects the groups on the system, it should only include the local and built-in group accounts and not domain group accounts. However, it is important to note that domain group accounts can still be looked up. Also, note that the subgroups of the group will not be resolved to find indirect user and group members. If the subgroups need to be resolved, it should be done using the sid_object. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a group_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 623: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< group_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.11
- Reason: Replaced by the group_sid_object. This object uses trustee names for identifying accounts on the system. Trustee names are not unique and the group_sid_object, which uses trustee SIDs which are unique, should be used instead. See the group_sid_object.
- Comment: This object has been deprecated and will be removed in version 6.0 of the language.

The group_object element is used by a group test to define the specific group(s) (identified by name) to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 624: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
group	oval-def:EntityObjectStringType (1..1)	The group element holds a string that represents the name of a particular group. In Windows, group names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, the group should be identified in the form: "domaingroup name". In a local environment, the group should be identified in the form: "computer namegroup name". If the group is a built-in group, the group should be identified in the form: "group name" without a domain component.
oval-def:filter	n/a (0..unbounded)	

< group_state > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.11
- Reason: Replaced by the group_sid_state. This state uses trustee names for identifying accounts on the system. Trustee names are not unique and the group_sid_state, which uses trustee SIDs which are unique, should be used instead. See the group_sid_state.
- Comment: This state has been deprecated and will be removed in version 6.0 of the language.

The group_state element enumerates the different users and subgroups directly associated with a Windows group. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 625: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
group	oval-def:EntityStateStringType (0..1)	The group element holds a string that represents the name of a particular group. In Windows, group names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, groups should be identified in the form: “domaingroup name”. For local groups use: “computer namegroup name”. For built-in accounts on the system, use the group name without a domain.
user	oval-def:EntityStateStringType (0..1)	The user element holds a string that represents the name of a particular user. In Windows, user names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, users should be identified in the form: “domainuser name”. For local users use: “computer nameuser name”. For built-in accounts on the system, use the user name without a domain.The user element can be included multiple times in a system characteristic item in order to record that a group contains a number of different users. Note that the entity_check attribute associated with EntityStateStringType guides the evaluation of entities like user that refer to items that can occur an unbounded number of times.
subgroup	oval-def:EntityStateStringType (0..1)	A string that represents the name of a particular subgroup in the specified group. In Windows, group names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, the subgroups should be identified in the form: “domaingroup name”. In a local environment, the subgroups should be identified in the form: “computer namegroup name”. If the subgroups are built-in groups, the subgroups should be identified in the form: “group name” without a domain component.The subgroup element can be included multiple times in a system characteristic item in order to record that a group contains a number of different subgroups. Note that the entity_check attribute associated with EntityStateStringType guides the evaluation of entities like the subgroup entity that refer to items that can occur an unbounded number of times.

< group_sid_test >

The group_sid_test allows the different users and subgroups, that directly belong to specific groups (identified by SID), to be tested. When the group_sid_test collects the group SIDs on the system, it should only include the local and built-in group SIDs and not domain group SIDs. However, it is important to note that domain group SIDs can still be looked up. Also, note that the subgroups of the group will not be resolved to find indirect user and group members. If the subgroups need to be resolved, it should be done using the sid_sid_object. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a group_sid_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 626: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< group_sid_object >

The group_sid_object element is used by a group_test to define the specific group(s) (identified by SID) to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 627: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
group_sid	oval-def:EntityObjectStringType (1..1)	The group_sid entity holds a string that represents the SID of a particular group.
oval-def:filter	n/a (0..unbounded)	

< group_sid_state >

The group_state element enumerates the different users and subgroups directly associated with a Windows group. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 628: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
group_sid	oval-def:EntityStateStringType (0..1)	The group_sid entity holds a string that represents the SID of a particular group.
user_sid	oval-def:EntityStateStringType (0..1)	The user_sid entity holds a string that represents the SID of a particular user. This entity can be included multiple times in a system characteristic item in order to record that a group contains a number of different users. Note that the entity_check attribute associated with EntityStateStringType guides the evaluation of entities like user that refer to items that can occur an unbounded number of times.
subgroup_sid	oval-def:EntityStateStringType (0..1)	The subgroup_sid entity holds a string that represents the SID of particular subgroup in the specified group. This entity can be included multiple times in a system characteristic item in order to record that a group contains a number of different subgroups. Note that the entity_check attribute associated with EntityStateStringType guides the evaluation of entities like subgroup_sid that refer to items that can occur an unbounded number of times.

< interface_test >

The interface test enumerate various attributes about the interfaces on a system. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an interface_object and the optional state element specifies the interface information to check.

Extends: oval-def:TestType

Child Elements

Table 629: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< interface_object >

The interface_object element is used by an interface test to define the specific interfaces(s) to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An interface object consists of a single name entity that identifies which interface is being specified. For help understanding this object, see the MIB_IFROW and MIB_IPADDRROW structures.

Extends: oval-def:ObjectType

Child Elements

Table 630: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityObjectStringType (1..1)	The name element specifies the name of an interface.
oval-def:filter	n/a (0..unbounded)	

< interface_state >

The interface_state element enumerates the different properties associate with a Windows interface. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 631: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	The name element specifies the name of an interface.
index	oval-def:EntityStateIntType (0..1)	The index element specifies index that identifies the interface.
type	win-def:EntityStateInterfaceTypeType (0..1)	The type element specifies the type of interface which is limited to certain set of values.
hardware_addr	oval-def:EntityStateStringType (0..1)	The hardware_addr entity is the hardware or MAC address of the physical network card. MAC addresses should be formatted according to the IEEE 802-2001 standard which states that a MAC address is a sequence of six octet values, separated by hyphens, where each octet is represented by two hexadecimal digits. Uppercase letters should also be used to represent the hexadecimal digits A through F.
inet_addr	oval-def:EntityStateIPAddressStringType (0..1)	The inet_addr element specifies the IP address. Note that the IP address can be IPv4 or IPv6. If the IP address is an IPv6 address, this entity will be expressed as an IPv6 address prefix using CIDR notation and the netmask entity will not be collected.
broadcast_addr	oval-def:EntityStateIPAddressStringType (0..1)	The broadcast_addr element specifies the broadcast address. A broadcast address is typically the IP address with the host portion set to either all zeros or all ones. Note that the IP address can be IPv4 or IPv6.
netmask	oval-def:EntityStateIPAddressStringType (0..1)	The netmask element specifies the subnet mask for the IP address. Note that if the inet_addr entity contains an IPv6 address prefix, this entity will not be collected.
addr_type	win-def:EntityStateAddrTypeType (0..1)	The addr_type element specifies the address type or state of a specific interface. Each interface can be associated with more than one value meaning the addr_type element can occur multiple times in a system characteristic item. Note that the entity_check attribute associated with EntityStateAddrTypeType guides the evaluation of unbounded entities like addr_type.

< junction_test >

The junction_test is used to obtain canonical path information for junctions (reparse points) on Windows filesystems.

Extends: oval-def:TestType

Child Elements

Table 632: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< junction_object >

The junction_object element is used by a junction_test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A junction_object consists of a path entity that contains the path to a symbolic link file. The resulting item identifies the canonical path of the link target (followed to its final destination, if there are intermediate links), an error if the link target does not exist or is a circular link (e.g., a link to itself). If the directory located at path is not a junction, or if there is no directory located at the path, then any resulting item would itself have a status of does not exist.

Extends: oval-def:ObjectType

Child Elements

Table 633: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	win-def:FileBehaviors (0..1)	
path	oval-def:EntityObjectStringType (1..1)	Specifies the path to the junction.
oval-def:filter	n/a (0..unbounded)	

< junction_state >

The junction_state element defines a value used to evaluate the result of a specific junction_object item.

Extends: oval-def:StateType

Child Elements

Table 634: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
path	oval-def:EntityStateStringType (0..1)	Specifies the path used to create the object.
canonical_path	oval-def:EntityStateStringType (0..1)	Specifies the canonical path for the target of a Windows junction specified by the path.
windows_view	win-def:EntityStateWindowsViewType (0..1)	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.

< license_test >

The license_test is used to check the content of a particular entry in the Windows registry HKLMSYSTEMCurrentControlSetControlProductOptions key, ProductPolicy value. Access to this data is exposed by the functions NtQueryLicenseValue (and also, in version 6.0 and higher, ZwQueryLicenseValue) in NTDLL.DLL.

Extends: oval-def:TestType

Child Elements

Table 635: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< license_object >

The license_object element is used by a license_test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 636: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityObjectStringType (1..1)	The name entity provides the address of a UNICODE_STRING structure for the name of the value for which data is desired, for example, TabletPCPlatformInput-core-EnableTouchUI.
oval-def:filter	n/a (0..unbounded)	

< license_state >

The license_state element defines the different information that can be found in the Windows license registry value. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 637: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	The name entity corresponds to the license_object name entity.
type	win-def:EntityStateRegistryTypeType (0..1)	The optional type entity provides the type of data that is expected: REG_SZ (0x01) for a string; REG_BINARY (0x03) for binary data; REG_DWORD (0x04) for a dword.
value	oval-def:EntityStateAnySimpleType (0..1)	The value entity allows a test to be written against the value held within the specified license entry(-ies). If the value being tested is of type REG_BINARY, then the datatype attribute should be set to 'binary' and the data represented by the value entity should follow the xsd:hexBinary form. (each binary octet is encoded as two hex digits) If the value being tested is of type REG_DWORD, then the datatype attribute should be set to 'int' and the value entity should represent the data as an integer. If the specified registry key is of type REG_SZ, then the datatype should be 'string' and the value entity should be a copy of the string.Note that if the intent is to test a version number held in the license entry (as a reg_sz) then instead of setting the datatype to 'string', the datatype can be set to 'version'. This allows tools performing the evaluation to know how to perform less than and greater than operations correctly.

< lockoutpolicy_test >

The lockout policy test enumerates various attributes associated with lockout information for users and global groups in the security database. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a lockoutpolicy_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 638: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< lockoutpolicy_object >

The lockoutpolicy_object element is used by a lockout policy test to define those objects to evaluated based on a specified state. There is actually only one object relating to lockout policy and this is the system as a whole. Therefore, there are no child entities defined. Any OVAL Test written to check lockout policy will reference the same lockoutpolicy_object which is basically an empty object element.

Extends: oval-def:ObjectType

< lockoutpolicy_state >

The lockoutpolicy_state element specifies the various attributes associated with lockout information for users and global groups in the security database. A lockout policy test will reference a specific instance of this state that defines the exact settings that need to be evaluated. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 639: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
force_logoff	oval-def:EntityStateIntType (0..1)	Specifies, in seconds (from a DWORD), the amount of time between the end of the valid logon time and the time when the user is forced to log off the network. A value of TIMEQ_FOREVER (max DWORD value, 4294967295) indicates that the user is never forced to log off. A value of zero indicates that the user will be forced to log off immediately when the valid logon time expires. See the USER_MODALS_INFO_0 structure returned by a call to NetUserModalsGet().
lockout_duration	oval-def:EntityStateIntType (0..1)	Specifies, in seconds, how long a locked account remains locked before it is automatically unlocked. See the USER_MODALS_INFO_3 structure returned by a call to NetUserModalsGet().
lockout_observation_window	oval-def:EntityStateIntType (0..1)	Specifies the maximum time, in seconds, that can elapse between any two failed logon attempts before lockout occurs. See the USER_MODALS_INFO_3 structure returned by a call to NetUserModalsGet().
lockout_threshold	oval-def:EntityStateIntType (0..1)	Specifies the number of invalid password authentications that can occur before an account is marked "locked out." See the USER_MODALS_INFO_3 structure returned by a call to NetUserModalsGet().

< metabase_test >

The metabase test is used to check information found in the Windows metabase. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a metabase_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 640: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< metabase_object >

The metabase_object element is used by a metabase test to define the specific metabase item(s) to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A metabase object defines the key and id of the item(s).

Extends: oval-def:ObjectType

Child Elements

Table 641: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
key	oval-def:EntityObjectStringType (1..1)	The key element specifies a metabase key.
id	oval-def:EntityObjectIntType (1..1)	The id element specifies a particular object under the metabase key. If the xsi:nil attribute is set to true, then the object being specified is the higher level key. In this case, the id element should not be collected or used in analysis. Setting xsi:nil equal to true is different than using a .* pattern match, says to collect every id under a given key. The most likely use for xsi:nil within a metabase object is when checking for the existence of a particular key, without regards to the different ids associated with it.
oval-def:filter	n/a (0..unbounded)	

< metabase_state >

The metabase_state element defines the different metadata associate with a metabase item. This includes the name, user type, data type, and the actual data. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 642: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
key	oval-def:EntityStateStringType (0..1)	The key element specifies a metabase key.
id	oval-def:EntityStateIntType (0..1)	The id element specifies a particular object under the metabase key.
name	oval-def:EntityStateStringType (0..1)	The name element describes the name of the specified metabase object. This is intended to be the string name of the constant from IIScnfg.h, e.g., MD_KEY_TYPE.
user_type	oval-def:EntityStateStringType (0..1)	The user_type element is an unsigned 32-bit integer (DWORD) that specifies the user type of the data. See the METADATA_RECORD structure.
data_type	oval-def:EntityStateStringType (0..1)	The data_type element identifies the type of data in the metabase entry. See the METADATA_RECORD structure.
data	oval-def:EntityStateAnySimpleType (0..1)	The actual data of the named item under the specified metabase key

---

< ntuser_test >

The ntuser test is used to check metadata associated with Windows ntuser.dat files. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a ntuser_object and the optional state element specifies the ntuser data to check.

Extends: oval-def:TestType

Child Elements

Table 643: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< ntuser_object >

The ntuser_object element is used to specify which metadata should be collected from a Windows ntuser.dat file. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 644: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	win-def:NTUserBehaviors (0..1)	
key	oval-def:EntityObjectStringType (1..1)	The key element describes a registry key to be collected. Note that the hive portion of the string should not be included, as this data is not neccessary for the ntuser test and would normally reside in the HKCU hive.
name	oval-def:EntityObjectStringType (1..1)	The name element describes the name assigned to a value associated with a specific registry key. If an empty string is specified for the name element, the registry key's default value should be collected. If the xsi:nil attribute is set to true, then the object being specified is the higher level key. In this case, the name element should not be collected or used in analysis. Setting xsi:nil equal to true on an element is different than using a .* pattern match. A .* pattern match says to collect every name under a given hive/key. The most likely use for xsi:nil within a registry object is when checking for the existence of a particular key, without regards to the different names associated with it.
oval-def:filter	n/a (0..unbounded)	

< ntuser_state >

The ntuser_state element defines the different metadata associated with a ntuser.dat file. This includes the key, name, type, and value. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 645: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
key	oval-def:EntityStateStringType (0..1)	This element describes a registry key normally found in the HKCU hive to be tested.
name	oval-def:EntityStateStringType (0..1)	This element describes the name of a value of a registry key.
sid	oval-def:EntityStateStringType (0..1)	This element holds a string that represents the SID of a particular user.
username	oval-def:EntityStateStringType (0..1)	The username entity holds a string that represents the name of a particular user. In Windows, user names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, users should be identified in the form: “domainuser name”. For local users use: “computer nameuser name”.
account_type	win-def:EntityStateNTUserAccountTypeType (0..1)	The account_type element describes if the user account is a local account or domain account.
logged_on	oval-def:EntityStateBoolType (0..1)	The logged_on element describes if the user account is currently logged on to the computer.
enabled	oval-def:EntityStateBoolType (0..1)	The enabled element describes if the user account is enabled or disabled.
date_modified	oval-def:EntityStateIntType (0..1)	Time of last modification of file. The integer should represent the FILETIME structure which is a 64-bit value representing the number of 100-nanosecond intervals since January 1, 1601 (UTC).
days_since_modified	oval-def:EntityStateIntType (0..1)	The number of days since the ntuser.dat file was last modified. The value should be rounded up to the next whole integer.
filepath	oval-def:EntityStateStringType (0..1)	This element describes the filepath of the ntuser.dat file.
last_write_time	oval-def:EntityStateIntType (0..1)	The last time that the key or any of its value entries was modified. The value of this entity represents the FILETIME structure which is a 64-bit value representing the number of 100-nanosecond intervals since January 1, 1601 (UTC). Last write time can be queried on a key or name. When collecting only information about a registry key the last write time will be the time the key or any of its entiries was written to. When collecting only information about a registry name the last write time will be the time the name was written to. See the RegQueryInfoKey function lpftLastWriteTime.
type	win-def:EntityStateRegistryTypeType (0..1)	The type entity allows a test to be written against the registy type associated with the specified registry key(s). Please refer to the documentation on the EntityStateRegistryTypeType for more information about the different valid individual types.
value	oval-def:EntityStateAnySimpleType (0..1)	The value entity allows a test to be written against the value held within the specified registry key(s). If the value being tested is of type REG_BINARY, then the datatype attribute should be set to ‘binary’ and the data represented by the value entity should follow the xsd:hexBinary form. (each binary octet is encoded as two hex digits) If the value being tested is of type REG_DWORD or REG_QWORD, then the datatype attribute should be set to ‘int’ and the value entity should represent the data as an integer. If the value being tested is of type REG_EXPAND_SZ, then the datatype attribute should be set to ‘string’ and the pre-expanded string should be represented by the value entity. If the value being tested is of type REG_MULTI_SZ, then only a single string (one of the multiple strings) should be tested using the value entity with the datatype attribute set to ‘string’. In order to test multiple values, multiple OVAL registry tests should be used. If the specified registry key is of type REG_SZ, then the datatype should be ‘string’ and the value entity should be a copy of the string.Note that if the intent is to test a version number held in the registry (as a reg_sz) then instead of setting the datatype

== NTUserBehaviors ==

The NTUserBehaviors complex type defines a number of behaviors that allow a more detailed definition of the ntuser_object being specified. 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.

Attributes

Table 646: Attributes

Attribute	Type	Desc.
include_default	xsd:boolean (optional *default*='false')	'include_default' defines if the Window's local Default ntuser.dat file is included in the results. By default, this file is not included in the results.

The Default User's directory which contains the ntuser.dat file is stored in the registry at 'HKEY_LOCAL_MACHINE/SOFTW

- - max_depth
  - Restriction of xsd:integer (optional *default*='-1')
  - 'max_depth' defines the maximum depth of recursion to perform when a recurse_direction is specified. A value of '0' is equivalent to no recursion, '1' means to step only one directory level up/down, and so on. The default value is '-1' meaning no limitation. For a 'max_depth' of -1 or any value of 1 or more the starting key must be considered in the recursive search.

Note that the default recurse_direction behavior is 'none' so even though max_depth specifies no limitation by default, the recurse_direction behavior turns recursion off. Note that this behavior only applies with the equality operation on the key entity.

- - recurse_direction
  - Restriction of xsd:string (optional *default*='none') ('none', 'up', 'down')
  - 'recurse_direction' defines the direction, either 'up' to parent keys, or 'down' into child keys to recursively search for registry keys. When recursing up or down, one is limited by the max_depth behavior. Note that it is not an error if max_depth specifies a certain level of recursion and that level does not exist. Recursing should only go as deep as available. The default value is 'none' for no recursion.

Note that this behavior only applies with the equality operation on the key entity.

- - windows_view
  - Restriction of xsd:string (optional *default*='64_bit') ('32_bit', '64_bit')
  - 64-bit versions of Windows provide an alternate file system and registry views to 32-bit applications. This behavior allows the OVAL Object to specify which view should be examined. This behavior only applies to 64-bit Windows, and must not be applied on other platforms.

Note that the values have the following meaning: '64_bit' – Indicates that the 64-bit view on 64-bit Windows operating systems must be examined. On a 32-bit system, the Object must be evaluated without applying the behavior. '32_bit' – Indicates that the 32-bit view must be examined. On a 32-bit system, the Object must be evaluated without applying the behavior. It is recommended that the corresponding 'windows_view' entity be set on the OVAL Items that are collected when this behavior is used to distinguish between the OVAL Items that are collected in the 32-bit or 64-bit views.

< passwordpolicy_test >

The password policy test is used to check specific policy associated with passwords. It is important to note that these policies are specific to certain versions of Windows. As a result, the documentation for that version of Windows should be consulted for more information. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a passwordpolicy_object and the optional state element specifies the metadata to check.

NOTE: This information is stored in the SAM or Active Directory but is encrypted or hidden so the registry_test and activedirectory57_test are of no use. If this can be figured out, then the password_policy test is not needed.

Extends: oval-def:TestType

Child Elements

Table 647: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< passwordpolicy_object >

The passwordpolicy_object element is used by a password policy test to define those objects to evaluated based on a specified state. There is actually only one object relating to password policy and this is the system as a whole. Therefore, there are no child entities defined. Any OVAL Test written to check password policy will reference the same passwordpolicy_object which is basically an empty object element.

Extends: oval-def:ObjectType

< passwordpolicy_state >

The passwordpolicy_state element specifies the various policies associated with passwords. A password policy test will reference a specific instance of this state that defines the exact settings that need to be evaluated.

Extends: oval-def:StateType

Child Elements

Table 648: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
max_passwd_age	oval-def:EntityStateIntType (0..1)	Specifies, in seconds (from a DWORD), the maximum allowable password age. A value of TIMEQ_FOREVER (max DWORD value, 4294967295) indicates that the password never expires. The minimum valid value for this element is ONE_DAY (86400). See the USER_MODALS_INFO_0 structure returned by a call to NetUserModalsGet().
min_passwd_age	oval-def:EntityStateIntType (0..1)	Specifies the minimum number of seconds that can elapse between the time a password changes and when it can be changed again. A value of zero indicates that no delay is required between password updates.
min_passwd_len	oval-def:EntityStateIntType (0..1)	Specifies the minimum allowable password length. Valid values for this element are zero through PWLEN.
password_hist_len	oval-def:EntityStateIntType (0..1)	Specifies the length of password history maintained. A new password cannot match any of the previous usrmod0_password_hist_len passwords. Valid values for this element are zero through DEF_MAX_PWHIST.
password_complexity	oval-def:EntityStateBoolType (0..1)	A boolean value that signifies whether passwords must meet the complexity requirements put forth by the operating system.
reversible_encryption	oval-def:EntityStateBoolType (0..1)	Determines whether or not passwords are stored using reversible encryption.
anonymous_name_lookup	oval-def:EntityStateBoolType (0..1)	Determines whether or not an anonymous user may query the local LSA policy.

< peheader_test >

The peheader_test is used to check data from a Portable Executable file header. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a peheader_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 649: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< peheader_object >

The peheader_object is used by a peheader_test to define the specific file(s) whose headers should be evaluated. The peheader_object will collect header information from PE files. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A peheader_object defines the path and filename or complete filepath of the file(s). In addition, a number of behaviors may be provided that help guide the collection of objects. Please refer to the PEHeaderBehaviors complex type for more information about specific behaviors.

The set of files whose headers should be evaluated may be identified with either a complete filepath or a path and filename. Only one of these options may be selected.

It is important to note that 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. This is because the ‘filepath’ element represents an absolute path to a particular file and it is not possible to recurse over a file.

Extends: oval-def:ObjectType

Child Elements

Table 650: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	win-def:FileBehaviors (0..1)	
filepath	oval-def:EntityObjectStringType (1..1)	The filepath element specifies the absolute path for a PE file on the machine. A directory cannot be specified as a filepath.
path	oval-def:EntityObjectStringType (1..1)	The path element specifies the directory component of the absolute path to a PE file on the machine.
filename	oval-def:EntityObjectStringType (1..1)	The filename element specifies the name of a PE file to evaluate.
oval-def:filter	n/a (0..unbounded)	

< peheader_state >

The peheader_state defines the different metadata associated with the header of a PE file. Please refer to the individual elements in the schema for more details about what each represents. For more information, please see the documentation for the IMAGE_FILE_HEADER and IMAGE_OPTIONAL_HEADER structures.

Extends: oval-def:StateType

Child Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-def:EntityStateStringType (0..1)	The filepath element specifies the absolu
path	oval-def:EntityStateStringType (0..1)	The path element specifies the directory
filename	oval-def:EntityStateStringType (0..1)	The filename element specifies the name
header_signature	oval-def:EntityStateStringType (0..1)	The header_signature entity is the signat
target_machine_type	win-def:EntityStatePeTargetMachineType (0..1)	The target_machine_type entity is an un
number_of_sections	oval-def:EntityStateIntType (0..1)	The number_of_sections entity is an uns
time_date_stamp	oval-def:EntityStateIntType (0..1)	The time_date_stamp entity is an unsign
pointer_to_symbol_table	oval-def:EntityStateIntType (0..1)	The pointer_to_symbol_table entity is ar
number_of_symbols	oval-def:EntityStateIntType (0..1)	The number_of_symbols entity is an uns
size_of_optional_header	oval-def:EntityStateIntType (0..1)	The size_of_optional_header entity is an
image_file_relocs_stripped	oval-def:EntityStateBoolType (0..1)	The image_file_relocs_stripped entity is
image_file_executable_image	oval-def:EntityStateBoolType (0..1)	The image_file_executable_image entity
image_file_line_nums_stripped	oval-def:EntityStateBoolType (0..1)	The image_file_line_nums_stripped enti
image_file_local_syms_stripped	oval-def:EntityStateBoolType (0..1)	The image_file_local_syms_stripped ent
image_file_aggresive_ws_trim	oval-def:EntityStateBoolType (0..1)	The image_file_aggressive_ws_trim enti
image_file_large_address_aware	oval-def:EntityStateBoolType (0..1)	The image_file_large_address_aware en
image_file_16bit_machine	oval-def:EntityStateBoolType (0..1)	The image_file_16bit_machine entity is
image_file_bytes_reversed_lo	oval-def:EntityStateBoolType (0..1)	The image_file_bytes_reversed_lo entity
image_file_32bit_machine	oval-def:EntityStateBoolType (0..1)	The image_file_32bit_machine entity is
image_file_debug_stripped	oval-def:EntityStateBoolType (0..1)	The image_file_debug_stripped entity is
image_file_removable_run_from_swap	oval-def:EntityStateBoolType (0..1)	The image_file_removable_run_from_sv
image_file_system	oval-def:EntityStateBoolType (0..1)	The image_file_system entity is a boolea
image_file_dll	oval-def:EntityStateBoolType (0..1)	The image_file_dll entity is a boolean va
image_file_up_system_only	oval-def:EntityStateBoolType (0..1)	The image_file_up_system_only entity i
image_file_bytes_reveresed_hi	oval-def:EntityStateBoolType (0..1)	The image_file_bytes_reversed_hi entity
magic_number	oval-def:EntityStateIntType (0..1)	The magic_number entity is an unsigned
major_linker_version	oval-def:EntityStateIntType (0..1)	The major_linker_version entity is a BY
minor_linker_version	oval-def:EntityStateIntType (0..1)	The minor_linker_version entity is a BY
size_of_code	oval-def:EntityStateIntType (0..1)	The size_of_code entity is an unsigned 3
size_of_initialized_data	oval-def:EntityStateIntType (0..1)	The size_of_initialized_data entity is an
size_of_uninitialized_data	oval-def:EntityStateIntType (0..1)	The size_of_uninitialized_data entity is a
address_of_entry_point	oval-def:EntityStateIntType (0..1)	The address_of_entry_point entity is an
base_of_code	oval-def:EntityStateIntType (0..1)	The base_of_code entity is an unsigned 3
base_of_data	oval-def:EntityStateIntType (0..1)	The base_of_data entity is an unsigned 3
image_base_address	oval-def:EntityStateIntType (0..1)	The image_base_address entity is an uns
section_alignment	oval-def:EntityStateIntType (0..1)	The section_alignment entity is an unsig
file_alignment	oval-def:EntityStateIntType (0..1)	The file_alignment entity is an unsigned
major_operating_system_version	oval-def:EntityStateIntType (0..1)	The major_operating_system_version en
minor_operating_system_version	oval-def:EntityStateIntType (0..1)	The minor_operating_system_version en
major_image_version	oval-def:EntityStateIntType (0..1)	The major_image_version entity is an un
minor_image_version	oval-def:EntityStateIntType (0..1)	The minor_image_version entity is an un
major_subsystem_version	oval-def:EntityStateIntType (0..1)	The major_subsystem_version entity is a
minor_susbsystem_version	oval-def:EntityStateIntType (0..1)	The minor_subsystem_version entity is a
size_of_image	oval-def:EntityStateIntType (0..1)	The size_of_image entity is an unsigned
size_of_headers	oval-def:EntityStateIntType (0..1)	The size_of_headers entity is an unsigne
checksum	oval-def:EntityStateIntType (0..1)	The checksum entity is an unsigned 32-b
subsystem	win-def:EntityStatePeSubsystemType (0..1)	The subsystem entity is an unsigned 32-
dll_characteristics	oval-def:EntityStateIntType (0..1)	The dll_characteristics entity is an unsig
size_of_stack_reserve	oval-def:EntityStateIntType (0..1)	The time_date_stamp entity is an unsign

Table

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
size_of_stack_commit	oval-def:EntityStateIntType (0..1)	The time_date_stamp entity is an unsign
size_of_heap_reserve	oval-def:EntityStateIntType (0..1)	The time_date_stamp entity is an unsign
size_of_heap_commit	oval-def:EntityStateIntType (0..1)	The time_date_stamp entity is an unsign
loader_flags	oval-def:EntityStateIntType (0..1)	The loader_flags entity is an unsigned 32
number_of_rva_and_sizes	oval-def:EntityStateIntType (0..1)	The number_of_rva_and_sizes entity is a
real_number_of_directory_entries	oval-def:EntityStateIntType (0..1)	The real_number_of_directory_entries e
windows_view	win-def:EntityStateWindowsViewType (0..1)	The windows view value to which this w

< port_test >

The port test is used to check information about the available ports on a Windows system. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a port_object and the optional state element specifies the port information to check.

Extends: oval-def:TestType

Child Elements

Table 652: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< port_object >

The port_object element is used by a port test to define the specific port(s) to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A port object defines the local address, port number, and protocol of the port(s).

Extends: oval-def:ObjectType

Child Elements

Table 653: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
local_address	oval-def:EntityObjectIPAddressStringType (1..1)	This element specifies the local IP address the listening port is bound to. Note that the IP address can be IPv4 or IPv6.
local_port	oval-def:EntityObjectIntType (1..1)	This element specifies the number assigned to the local listening port.
protocol	win-def:EntityObjectProtocolType (1..1)	This element specifies the type of listening port. It is restricted to either TCP or UDP.
oval-def:filter	n/a (0..unbounded)	

< port_state >

The port_state element defines the different metadata associate with a Windows port. This includes the local address, port number, protocol, and pid. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 654: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
local_address	oval-def:EntityStateIPAddressStringType (0..1)	This element specifies the local IP address the listening port is bound to. Note that the IP address can be IPv4 or IPv6.
local_port	oval-def:EntityStateIntType (0..1)	This element specifies the number assigned to the local listening port.
protocol	win-def:EntityStateProtocolType (0..1)	This element specifies the type of listening port. It is restricted to either TCP or UDP.
pid	oval-def:EntityStateIntType (0..1)	The id given to the process that is associated with the specified listening port.
foreign_address	oval-def:EntityStateIPAddressStringType (0..1)	This is the IP address with which the program is communicating, or with which it will communicate, in the case of a listening server. Note that the IP address can be IPv4 or IPv6.
foreign_port	oval-def:EntityStateStringType (0..1)	This is the TCP or UDP port to which the program communicates. In the case of a listening program accepting new connections, this is usually ‘0’.

< printereffectiverights_test >

The printer effective rights test is used to check the effective rights associated with Windows printers. The printereffectiverights_test element extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a printereffectiverights_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 655: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< printereffectiverights_object >

Extends: oval-def:ObjectType

Child Elements

Table 656: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	win-def:PrinterEffectiveRightsBehaviors (0..1)	
printer_name	oval-def:EntityObjectStringType (1..1)	The printer_name element describes a printer that a user may have rights on.
trustee_sid	oval-def:EntityObjectStringType (1..1)	The trustee_sid entity identifies a unique SID associated with a user, group, system, or program (such as a Windows service). If an operation other than equals is used to identify matching trustees (i.e. not equal, or a pattern match) then the resulting matches shall be limited to only the trustees referenced in the printer's Security Descriptor. The scope is limited here to ensure that it is possible to avoid unnecessarily resource intensive searches for trustees. Note that the larger scope of all known trustees may be obtained through the use of variables.
oval-def:filter	n/a (0..unbounded)	

< printereffectiverights_state >

The printereffectiverights_state element defines the different rights that can be associated with a given printereffectiverights_object. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 657: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
printer_name	oval-def:EntityStateStringType (0..1)	This element specifies the name of the printer.
trustee_sid	oval-def:EntityStateStringType (0..1)	The trustee_sid element is the unique SID that associated a user, group, system, or program (such as a Windows service).
standard_delete	oval-def:EntityStateBoolType (0..1)	The right to delete the object.
standard_read_control	oval-def:EntityStateBoolType (0..1)	The right to read the information in the object's Security Descriptor, not including the information in the SACL.
standard_write_dac	oval-def:EntityStateBoolType (0..1)	The right to modify the DACL in the object's Security Descriptor.
standard_write_owner	oval-def:EntityStateBoolType (0..1)	The right to change the owner in the object's Security Descriptor.
standard_synchronize	oval-def:EntityStateBoolType (0..1)	The right to use the object for synchronization. This enables a thread to wait until the object is in the signaled state. Some object types do not support this access right.
access_system_security	oval-def:EntityStateBoolType (0..1)	Indicates access to a system access control list (SACL).
generic_read	oval-def:EntityStateBoolType (0..1)	Read access.
generic_write	oval-def:EntityStateBoolType (0..1)	Write access.
generic_execute	oval-def:EntityStateBoolType (0..1)	Execute access.
generic_all	oval-def:EntityStateBoolType (0..1)	Read, write, and execute access.
printer_access_administer	oval-def:EntityStateBoolType (0..1)	
printer_access_use	oval-def:EntityStateBoolType (0..1)	
job_access_administer	oval-def:EntityStateBoolType (0..1)	
job_access_read	oval-def:EntityStateBoolType (0..1)	

== PrinterEffectiveRightsBehaviors ==

The PrinterEffectiveRightsBehaviors complex type defines a number of behaviors that allow a more detailed definition of the pritnereffectiverights_object being specified. 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.

Attributes

Table 658: Attributes

Attribute	Type	Desc.
include_group (Deprecated)	xsd:boolean (optional *default*='true')	'include_group' defines whether the group trustee name should be included in the object when the object is defined by a group trustee name. For example, the intent of an object defined by a group trustee name might be to retrieve all the user trustee names that are members of the group, but not the group trustee name itself.
resolve_group (Deprecated)	xsd:boolean (optional *default*='false')	The 'resolve_group' behavior defines whether an object set defined by a group SID should be resolved to return a set that contains all the user SIDs that are a member of that group. Note that all child groups should also be resolved any valid domain users that are members of the group should also be included. The intent of this behavior is to end up with a list of all individual users from that system that make up the group once everything has been resolved.

< process_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.8
- Reason: The process_test has been deprecated and replaced by the process58_test. The command line of a process cannot be used to uniquely identify a process. As a result, the pid entity was added to the process58_object. Please see the process58_test for additional information.

The process_test is used to check information found in the Windows processes. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a process_object and the optional state element references a process_state element that specifies the process information to check.

Extends: oval-def:TestType

Child Elements

Table 659: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< process_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.8
- Reason: The process_object has been deprecated and replaced by the process58_object. The command line of a process cannot be used to uniquely identify a process. As a result, the pid entity was added to the process58_object. Please see the process58_object for additional information.

The process_object element is used by a process test to define the specific process(es) to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A process_object defines the command line used to start the process(es).

Extends: oval-def:ObjectType

Child Elements

Table 660: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
command_line	oval-def:EntityObjectStringType (1..1)	The command_line entity is the string used to start the process. This includes any parameters that are part of the command line.

< process_state > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.8
- Reason: The process_state has been deprecated and replaced by the process58_state. The command line of a process cannot be used to uniquely identify a process. As a result, the pid entity was added to the process58_object. Please see the process58_state for additional information.

The process_state element defines the different metadata associate with a Windows process. This includes the command line, pid, ppid, image path, and current directory. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 661: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
command_line	oval-def:EntityStateStringType (0..1)	The command_line entity is the string used to start the process. This includes any parameters that are part of the command line.
pid	oval-def:EntityStateIntType (0..1)	The id given to the process that is created for a specified command line.
ppid	oval-def:EntityStateIntType (0..1)	The id given to the parent of the process that is created for the specified command line
priority	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	The base priority of the process. The priority value range is from 0 to 31.
image_path	oval-def:EntityStateStringType (0..1)	The image_path entity contains the name of the executable file in question.
current_dir	oval-def:EntityStateStringType (0..1)	The current_directory entity represents the current path to the executable.

< process58_test >

The process58_test is used to check information found in the Windows processes. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a process58_object and the optional state element references a process58_state element that specifies the process information to check.

Extends: oval-def:TestType

Child Elements

Table 662: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< process58_object >

The process58_object element is used by a process58_test to define the specific process(es) to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A process58_object defines the command line used to start the process(es)and pid.

Extends: oval-def:ObjectType

Child Elements

Table 663: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
command_line	oval-def:EntityObjectStringType (1..1)	The command_line entity is the string used to start the process. This includes any parameters that are part of the command line. Use xsi:nil='true' to disregard (and permit processes with non-existent commane_lines, such as the System process).
pid	oval-def:EntityObjectIntType (1..1)	The id given to the process that is created for a specified command line.
oval-def:filter	n/a (0..unbounded)	

< process58_state >

The process58_state element defines the different metadata associate with a Windows process. This includes the command line, pid, ppid, image path, and current directory. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 664: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
command_line	oval-def:EntityStateStringType (0..1)	The command_line entity is the string used to start the process. This includes any parameters that are part of the command line.
pid	oval-def:EntityStateIntType (0..1)	The id given to the process that is created for a specified command line.
ppid	oval-def:EntityStateIntType (0..1)	The id given to the parent of the process that is created for the specified command line
priority	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	The base priority of the process. The priority value range is from 0 to 31.
image_path	oval-def:EntityStateStringType (0..1)	The image_path entity represents the name of the executable file for the process.
current_dir	oval-def:EntityStateStringType (0..1)	The current_dir entity represents the current path to the executable file for the process.
creation_time	oval-def:EntityStateIntType (0..1)	The creation_time entity represents the creation time of the process. The value of this entity represents the FILETIME structure which is a 64-bit value representing the number of 100-nanosecond intervals since January 1, 1601 (UTC). See the GetProcessTimes function lpCreationTime.
dep_enabled	oval-def:EntityStateBoolType (0..1)	The dep_enabled entity represents whether or not data execution prevention (DEP) is enabled. See the GetProcessDEPPolicy lpFlags.
primary_window_text	oval-def:EntityStateStringType (0..1)	The primary_window_text entity represents the title of the primary window of the process. See the GetWindowText function.
name	oval-def:EntityStateStringType (0..1)	The name of the process.

< registry_test >

The registry test is used to check metadata associated with Windows registry key. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a registry_object and the optional state element specifies the registry data to check.

Extends: oval-def:TestType

Child Elements

Table 665: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< registry_object >

Extends: oval-def:ObjectType

Child Elements

Table 666: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	win-def:RegistryBehaviors (0..1)	
hive	win-def:EntityObjectRegistryHiveType (1..1)	The hive that the registry key belongs to. This is restricted to a specific set of values: HKEY_CLASSES_ROOT, HKEY_CURRENT_CONFIG, HKEY_CURRENT_USER, HKEY_CURRENT_USER_LOCAL_SETTINGS, HKEY_LOCAL_MACHINE, and HKEY_USERS.
key	oval-def:EntityObjectStringType (1..1)	The key element describes a registry key to be collected. Note that the hive portion of the string should not be included, as this data should be found under the hive element. If the xsi:nil attribute is set to true, then the object being specified is the higher level hive. In this case, the key element should not be collected or used in analysis. Setting xsi:nil equal to true is different than using a .* pattern match. A .* pattern match says to collect every key under a given hive.
name	oval-def:EntityObjectStringType (1..1)	The name element describes the name assigned to a value associated with a specific registry key. If an empty string is specified for the name element, the registry key's default value should be collected. If the xsi:nil attribute is set to true, then the object being specified is the higher level hive/key. In this case, the name element should not be collected or used in analysis. Setting xsi:nil equal to true on an element is different than using a .* pattern match. A .* pattern match says to collect every name under a given hive/key. The most likely use for xsi:nil within a registry object is when checking for the existence of a particular key, without regards to the different names associated with it.
oval-def:filter	n/a (0..unbounded)	

< registry_state >

The registry_state element defines the different metadata associate with a Windows registry key. This includes the hive, key, name, type, and value. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 667: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
hive	win-def:EntityStateRegistryHiveType (0..1)	The hive that the registry key belongs to. This is restricted to a specific set of values: HKEY_CLASSES_ROOT, HKEY_CURRENT_CONFIG, HKEY_CURRENT_USER, HKEY_CURRENT_USER_LOCAL_SETTINGS,HKEY_LOCAL_MACHINE, and HKEY_USERS.
key	oval-def:EntityStateStringType (0..1)	This element describes a registry key to be tested. Note that the hive portion of the string should not be included, as this data should be found under the hive element.
name	oval-def:EntityStateStringType (0..1)	This element describes the name of a value of a registry key. If the xsi:nil attribute is set to true, then the name element should not be used in analysis.
last_write_time	oval-def:EntityStateIntType (0..1)	The last time that the key or any of its value entries were modified. The value of this entity represents the FILETIME structure which is a 64-bit value representing the number of 100-nanosecond intervals since January 1, 1601 (UTC). Last write time can be queried on any key, with hives being classified as a type of key. When collecting only information about a registry hive or key the last write time will be the time the key or any of its entries were modified. When collecting only information about a registry name the last write time will be the time the containing key was modified. Thus when collecting information about a registry name, the last write time does not correlate directly to the specified name. See the RegQueryInfoKey function lpftLastWriteTime.
type	win-def:EntityStateRegistryTypeType (0..1)	The type entity allows a test to be written against the registy type associated with the specified registry key(s). Please refer to the documentation on the EntityStateRegistryTypeType for more information about the different valid individual types.
value	oval-def:EntityStateAnySimpleType (0..1)	The value entity allows a test to be written against the value held within the specified registry key(s). If the value being tested is of type REG_BINARY, then the datatype attribute should be set to 'binary' and the data represented by the value entity should follow the xsd:hexBinary form. (each binary octet is encoded as two hex digits) If the value being tested is of type REG_DWORD, REG_QWORD, REG_DWORD_LITTLE_ENDIAN, REG_DWORD_BIG_ENDIAN, and REG_QWORD_LITTLE_ENDIAN then the datatype attribute should be set to 'int' and the value entity should represent the data as an unsigned integer. DWORD and QWORD values represnt unsigned 32-bit and 64-bit integers, respectively. If the value being tested is of type REG_EXPAND_SZ, then the datatype attribute should be set to 'string' and the pre-expanded string should be represented by the value entity. If the value being tested is of type REG_MULTI_SZ, then only a single string (one of the multiple strings) should be tested using the value entity with the datatype attribute set to 'string'. In order to test multiple values, multiple OVAL registry tests should be used. If the specified registry key is of type REG_SZ, then the datatype should be 'string' and the value entity should be a copy of the string. If the value being tested is of type REG_LINK, then the datatype attribute should be set to 'string' and the null-terminated Unicode string should be represented by the value entity.Note that if the intent is to test a version number held in the registry (as a reg_sz) then instead of setting the datatype to 'string', the datatype can be set to 'version'. This allows tools performing the evaluation to know how to perform less than and greater than operations correctly.
expanded_value	oval-def:EntityStateAnySimpleType (0..1)	For registry values of type REG_EXPAND_SZ, this entity contains the expanded value. Otherwise, it should not exist.
windows_view	win-def:EntityStateWindowsViewType (0..1)	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.

== RegistryBehaviors ==

The RegistryBehaviors complex type defines a number of behaviors that allow a more detailed definition of the registry_object being specified. 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.

Attributes

Table 668: Attributes

Attribute	Type	Desc.
max_depth	Restriction of xsd:integer (optional *default*=’-1’)	‘max_depth’ defines the maximum depth of recursion to perform when a recurse_direction is specified. A value of ‘0’ is equivalent to no recursion, ‘1’ means to step only one directory level up/down, and so on. The default value is ‘-1’ meaning no limitation. For a ‘max_depth’ of -1 or any value of 1 or more the starting key must be considered in the recursive search.

Note that the default recurse_direction behavior is ‘none’ so even though max_depth specifies no limitation by default, the recurse_direction behavior turns recursion off. Note that this behavior only applies with the equality operation on the key entity.

- - recurse_direction
  - Restriction of xsd:string (optional *default*=’none’) (‘none’, ‘up’, ‘down’)
  - ‘recurse_direction’ defines the direction, either ‘up’ to parent keys, or ‘down’ into child keys to recursively search for registry keys. When recursing up or down, one is limited by the max_depth behavior. Note that it is not an error if max_depth specifies a certain level of recursion and that level does not exist. Recursing should only go as deep as available. The default value is ‘none’ for no recursion.

Note that this behavior only applies with the equality operation on the key entity.

- - windows_view
  - Restriction of xsd:string (optional *default*=’64_bit’) (‘32_bit’, ‘64_bit’)
  - 64-bit versions of Windows provide an alternate file system and registry views to 32-bit applications. This behavior allows the OVAL Object to specify which view should be examined. This behavior only applies to 64-bit Windows, and must not be applied on other platforms.

Note that the values have the following meaning: ‘64_bit’ - Indicates that the 64-bit view on 64-bit Windows operating systems must be examined. On a 32-bit system, the Object must be evaluated without applying the behavior. ‘32_bit’ - Indicates that the 32-bit view must be examined. On a 32-bit system, the Object must be evaluated without applying the behavior. It is recommended that the corresponding ‘windows_view’ entity be set on the OVAL Items that are collected when this behavior is used to distinguish between the OVAL Items that are collected in the 32-bit or 64-bit views.

< regkeyauditedpermissions53_test >

The registry key audited permissions test is used to check the audit permissions associated with Windows registry keys. Note that the trustee’s audited permissions are the audit permissons that the SACL grants to the trustee or to any groups of which the trustee is a member. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a regkeyauditedpermissions53_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 669: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< regkeyauditedpermissions53_object >

The regkeyauditedpermissions53_object element is used by a registry key audited permissions test to define the objects used to evalutate against the specified state. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic.

A regkeyauditedpermissions53_object is defined as a combination of a Windows registry key and trustee name. The hive and key elements represents the registry key to be evaluated while the trustee name represents the account (SID) to check audited permissions of. If multiple keys or SIDs are matched by either reference, then each possible combination of registry key and SID is a matching registry key audited permissions object. In addition, a number of behaviors may be provided that help guide the collection of objects. Please refer to the RegkeyAuditPermissions53Behaviors complex type for more information about specific behaviors.

Extends: oval-def:ObjectType

Child Elements

Table 670: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	win-def:RegkeyAuditPermissions53Behaviors (0..1)	
hive	win-def:EntityObjectRegistryHiveType (1..1)	The hive that the registry key belongs to. This is restricted to a specific set of values: HKEY_CLASSES_ROOT, HKEY_CURRENT_CONFIG, HKEY_CURRENT_USER, HKEY_CURRENT_USER_LOCAL_SETTINGS, HKEY_LOCAL_MACHINE, and HKEY_USERS.
key	oval-def:EntityObjectStringType (1..1)	The key element describes a registry key to be collected. Note that the hive portion of the string should not be included, as this data should be found under the hive element. If the xsi:nil attribute is set to true, then the object being specified is the higher level hive. In this case, the key element should not be collected or used in analysis. Setting xsi:nil equal to true is different than using a .* pattern match. A .* pattern match says to collect every key under a given hive.
trustee_sid	oval-def:EntityObjectStringType (1..1)	The trustee_sid entity identifies a unique SID associated with a user, group, system, or program (such as a Windows service). If an operation other than equals is used to identify matching trustees (i.e. not equal, or a pattern match) then the resulting matches shall be limited to only the trustees referenced in the registry key's Security Descriptor. The scope is limited here to avoid unnecessarily resource intensive searches for trustees. Note that the larger scope of all known trustees may be obtained through the use of variables.
oval-def:filter	n/a (0..unbounded)	

< regkeyauditedpermissions53_state >

The regkeyauditedpermissions53_state element defines the different audit permissions that can be associated with a given regkeyauditedpermissions53_object. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 671: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
hive	win-def:EntityStateRegistryHiveType (0..1)	This element specifies the hive of a registry key on the machine from which to retrieve the SACL.
key	oval-def:EntityStateStringType (0..1)	This element specifies a registry key on the machine from which to retrieve the SACL. Note that the hive portion of the string should not be inclueded, as this data should be found under the hive element.
trustee_sid	oval-def:EntityStateStringType (0..1)	The trustee_sid element is the unique SID that associated a user, group, system, or program (such as a Windows service).
standard_delete	win-def:EntityStateAuditType (0..1)	The right to delete the object.
standard_read_control	win-def:EntityStateAuditType (0..1)	The right to read the information in the object's Security Descriptor, not including the information in the SACL.
standard_write_dac	win-def:EntityStateAuditType (0..1)	The right to modify the DACL in the object's Security Descriptor.
standard_write_owner	win-def:EntityStateAuditType (0..1)	The right to change the owner in the object's Security Descriptor.
standard_synchronize (Deprecated)	win-def:EntityStateAuditType (0..1)	The right to use the object for synchronization. This enables a thread to wait until the object is in the signaled state. Some object types do not support this access right.
access_system_security	win-def:EntityStateAuditType (0..1)	Indicates access to a system access control list (SACL).
generic_read	win-def:EntityStateAuditType (0..1)	Read access.
generic_write	win-def:EntityStateAuditType (0..1)	Write access.
generic_execute	win-def:EntityStateAuditType (0..1)	Execute access.
generic_all	win-def:EntityStateAuditType (0..1)	Read, write, and execute access.
key_query_value	win-def:EntityStateAuditType (0..1)	
key_set_value	win-def:EntityStateAuditType (0..1)	
key_create_sub_key	win-def:EntityStateAuditType (0..1)	
key_enumerate_sub_keys	win-def:EntityStateAuditType (0..1)	
key_notify	win-def:EntityStateAuditType (0..1)	

== RegkeyAuditPermissions53Behaviors ==

The RegkeyAuditPermissions53Behaviors complex type defines a number of behaviors that allow a more detailed definition of the registrykeyauditedpermissions53_object being specified. 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.

The RegkeyAuditPermissions53Behaviors extend the win-def:RegistryBehaviors and therefore include the behaviors defined by that type.

Extends: win-def:RegistryBehaviors

Attributes

Table 672: Attributes

Attribute	Type	Desc.
include_group (Deprecated)	xsd:boolean (optional *default*='true')	'include_group' defines whether the group SID should be included in the object when the object is defined by a group SID. For example, the intent of an object defined by a group SID might be to retrieve all the user SIDs that are a member of the group, but not the group SID itself.
resolve_group (Deprecated)	xsd:boolean (optional *default*='false')	The 'resolve_group' behavior defines whether an object set defined by a group SID should be resolved to return a set that contains all the user SIDs that are a member of that group. Note that all child groups should also be resolved any valid domain users that are members of the group should also be included. The intent of this behavior is to end up with a list of all individual users from that system that make up the group once everything has been resolved.

< regkeyauditedpermissions_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.3
- Reason: Replaced by the regkeyauditedpermissions53_test. This test uses a trustee_name element for identifying trustees. Trustee names are not unique, and a new test was created to use trustee SIDs, which are unique. See the regkeyauditedpermissions53_test.
- Comment: This test has been deprecated and will be removed in version 6.0 of the language.

The registry key audited permissions test is used to check the audit permissions associated with Windows registry keys. Note that the trustee's audited permissions are the audit permissons that the SACL grants to the trustee or to any groups of which the trustee is a member. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a regkeyauditedpermissions_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 673: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< regkeyauditedpermissions_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.3
- Reason: Replaced by the regkeyauditedpermissions53_object. This object uses a trustee_name element for identifying trustees. Trustee names are not unique, and a new object was created to use trustee SIDs, which are unique. See the regkeyauditedpermissions53_object.
- Comment: This object has been deprecated and will be removed in version 6.0 of the language.

The regkeyauditedpermissions_object element is used by a registry key audited permissions test to define the objects used to evalutate against the specified state. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic.

A regkeyauditedpermissions_object is defined as a combination of a Windows registry key and trustee name. The hive and key elements represents the registry key to be evaluated while the trustee name represents the account (SID) to check audited permissions of. If multiple keys or SIDs are matched by either reference, then each possible combination of file and SID is a matching file audited permissions object. In addition, a number of behaviors may be provided that help guide the collection of objects. Please refer to the RegkeyAuditPermissionsBehaviors complex type for more information about specific behaviors.

Extends: oval-def:ObjectType

Child Elements

Table 674: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	win-def:RegkeyAuditPermissionsBehaviors (0..1)	
hive	win-def:EntityObjectRegistryHiveType (1..1)	The hive that the registry key belongs to. This is restricted to a specific set of values: HKEY_CLASSES_ROOT, HKEY_CURRENT_CONFIG, HKEY_CURRENT_USER, HKEY_CURRENT_USER_LOCAL_SETTINGS, HKEY_LOCAL_MACHINE, and HKEY_USERS.
key	oval-def:EntityObjectStringType (1..1)	The key element describes a registry key to be collected. Note that the hive portion of the string should not be included, as this data should be found under the hive element.
trustee_name	oval-def:EntityObjectStringType (1..1)	The trustee_name element is the unique name that associated a particular SID. A SID can be associated with a user, group, or program (such as a Windows service). In Windows, trustee names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, trustee names should be identified in the form: “domaintrustee name”. For local trustee names use: “computer nametrustee name”. For built-in accounts on the system, use the trustee name without a domain.

< regkeyauditedpermissions_state > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.3
- Reason: Replaced by the regkeyauditedpermissions53_state. This state uses a trustee_name element for identifying trustees. Trustee names are not unique, and a new state was created to use trustee SIDs, which are unique. See the regkeyauditedpermissions53_state.
- Comment: This state has been deprecated and will be removed in version 6.0 of the language.

The regkeyauditedpermissions_state element defines the different audit permissions that can be associated with a given regkeyauditedpermissions_object. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 675: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
hive	win-def:EntityStateRegistryHiveType (0..1)	This element specifies the hive of a registry key on the machine from which to retrieve the SACL.
key	oval-def:EntityStateStringType (0..1)	This element specifies a registry key on the machine from which to retrieve the SACL. Note that the hive portion of the string should not be inclueded, as this data should be found under the hive element.
trustee_name	oval-def:EntityStateStringType (0..1)	The unique name associated with a particular security identifier (SID). In Windows, trustee names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, trustee names should be identified in the form: "domaintrustee name". For local trustee names use: "computer nametrustee name". For built-in accounts on the system, use the trustee name without a domain.
standard_delete	win-def:EntityStateAuditType (0..1)	The right to delete the object.
standard_read_control	win-def:EntityStateAuditType (0..1)	The right to read the information in the object's Security Descriptor, not including the information in the SACL.
standard_write_dac	win-def:EntityStateAuditType (0..1)	The right to modify the DACL in the object's Security Descriptor.
standard_write_owner	win-def:EntityStateAuditType (0..1)	The right to change the owner in the object's Security Descriptor.
standard_synchronize	win-def:EntityStateAuditType (0..1)	The right to use the object for synchronization. This enables a thread to wait until the object is in the signaled state. Some object types do not support this access right.
access_system_security	win-def:EntityStateAuditType (0..1)	Indicates access to a system access control list (SACL).
generic_read	win-def:EntityStateAuditType (0..1)	Read access.
generic_write	win-def:EntityStateAuditType (0..1)	Write access.
generic_execute	win-def:EntityStateAuditType (0..1)	Execute access.
generic_all	win-def:EntityStateAuditType (0..1)	Read, write, and execute access.
key_query_value	win-def:EntityStateAuditType (0..1)	
key_set_value	win-def:EntityStateAuditType (0..1)	
key_create_sub_key	win-def:EntityStateAuditType (0..1)	
key_enumerate_sub_keys	win-def:EntityStateAuditType (0..1)	

== RegkeyAuditPermissionsBehaviors == (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.3
- Reason: Replaced by the RegkeyAuditPermissionsBehaviors53. The RegkeyAuditPermissionsBehaviors complex type is used by the regkeyauditedpermissions_test which uses a trustee_name element for identifying trustees. Trustee names are not unique, and a new test was created to use trustee SIDs, which are unique. This new test utilizes the RegkeyAuditPermissionsBehaviors53 complex type, and as a result, the RegkeyAuditPermissionsBehaviors complex type is no longer needed.
- Comment: This complex type has been deprecated and will be removed in version 6.0 of the language.

The RegkeyAuditPermissionsBehaviors complex type defines a number of behaviors that allow a more detailed definition of the registrykeyauditedpermissions_object being specified. 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.

The RegkeyAuditPermissionsBehaviors extend the win-def:RegistryBehaviors and therefore include the behaviors defined by that type.

Extends: win-def:RegistryBehaviors

Attributes

Table 676: Attributes

Attribute	Type	Desc.
include_group (Deprecated)	xsd:boolean (optional *default*=’true’)	‘include_group’ defines whether the group trustee name should be included in the object when the object is defined by a group trustee name. For example, the intent of an object defined by a group trustee name might be to retrieve all the user trustee names that are members of the group, but not the group trustee name itself.
resolve_group (Deprecated)	xsd:boolean (optional *default*=’false’)	The ‘resolve_group’ behavior defines whether an object set defined by a group SID should be resolved to return a set that contains all the user SIDs that are a member of that group. Note that all child groups should also be resolved any valid domain users that are members of the group should also be included. The intent of this behavior is to end up with a list of all individual users from that system that make up the group once everything has been resolved.

< regkeyeffectiverights53_test >

The registry key effective rights test is used to check the effective rights associated with Windows files. Note that the trustee’s effective access rights are the access rights that the DACL grants to the trustee or to any groups of which the trustee is a member. The regkeyeffectiverights53_test element extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a regkeyeffectiverights53_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 677: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< regkeyeffectiverights53_object >

The regkeyeffectiverights53_object element is used by a registry key effective rights test to define the objects used to evalutate against the specified state. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic.

A regkeyeffectiverights53_object is defined as a combination of a Windows registry and trustee SID. The key entity represents the registry key to be evaluated while the trustee SID represents the account (SID) to check effective rights of. If multiple files or SIDs are matched by either reference, then each possible combination of registry key and SID is a matching registry key effective rights object. In addition, a number of behaviors may be provided that help guide the collection of objects. Please refer to the RegkeyEffectiveRights53Behaviors complex type for more information about specific behaviors.

Extends: oval-def:ObjectType

Child Elements

Table 678: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	win-def:RegkeyEffectiveRights53Behaviors (0..1)	
hive	win-def:EntityObjectRegistryHiveType (1..1)	The hive that the registry key belongs to. This is restricted to a specific set of values: HKEY_CLASSES_ROOT, HKEY_CURRENT_CONFIG, HKEY_CURRENT_USER, HKEY_CURRENT_USER_LOCAL_SETTINGS,HKEY_LOCAL_MACHINE, and HKEY_USERS.
key	oval-def:EntityObjectStringType (1..1)	The key element describes a registry key to be collected. Note that the hive portion of the string should not be included, as this data should be found under the hive element. If the xsi:nil attribute is set to true, then the object being specified is the higher level hive. In this case, the key element should not be collected or used in analysis. Setting xsi:nil equal to true is different than using a .* pattern match. A .* pattern match says to collect every key under a given hive.
trustee_sid	oval-def:EntityObjectStringType (1..1)	The trustee_sid entity identifies a unique SID associated with a user, group, system, or program (such as a Windows service). If an operation other than equals is used to identify matching trustees (i.e. not equal, or a pattern match) then the resulting matches shall be limited to only the trustees referenced in the registry key's Security Descriptor. The scope is limited here to avoid unnecessarily resource intensive searches for trustees. Note that the larger scope of all known trustees may be obtained through the use of variables.
oval-def:filter	n/a (0..unbounded)	

< regkeyeffectiverights53_state >

The regkeyeffectiverights53_state element defines the different rights that can be associated with a given regkeyeffectiverights53_object. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 679: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
hive	win-def:EntityStateRegistryHiveType (0..1)	This element specifies the hive of a registry key on the machine from which to retrieve the SACL.
key	oval-def:EntityStateStringType (0..1)	This element specifies a registry key on the machine from which to retrieve the SACL. Note that the hive portion of the string should not be inclueded, as this data should be found under the hive element.
trustee_sid	oval-def:EntityStateStringType (0..1)	The trustee_sid element is the unique SID that associated a user, group, system, or program (such as a Windows service).
standard_delete	oval-def:EntityStateBoolType (0..1)	The right to delete the object.
standard_read_control	oval-def:EntityStateBoolType (0..1)	The right to read the information in the object's Security Descriptor, not including the information in the SACL.
standard_write_dac	oval-def:EntityStateBoolType (0..1)	The right to modify the DACL in the object's Security Descriptor.
standard_write_owner	oval-def:EntityStateBoolType (0..1)	The right to change the owner in the object's Security Descriptor.
standard_synchronize (Deprecated)	oval-def:EntityStateBoolType (0..1)	The right to use the object for synchronization. This enables a thread to wait until the object is in the signaled state. Some object types do not support this access right.
access_system_security	oval-def:EntityStateBoolType (0..1)	Indicates access to a system access control list (SACL).
generic_read	oval-def:EntityStateBoolType (0..1)	Read access.
generic_write	oval-def:EntityStateBoolType (0..1)	Write access.
generic_execute	oval-def:EntityStateBoolType (0..1)	Execute access.
generic_all	oval-def:EntityStateBoolType (0..1)	Read, write, and execute access.
key_query_value	oval-def:EntityStateBoolType (0..1)	
key_set_value	oval-def:EntityStateBoolType (0..1)	
key_create_sub_key	oval-def:EntityStateBoolType (0..1)	
key_enumerate_sub_keys	oval-def:EntityStateBoolType (0..1)	
key_notify	oval-def:EntityStateBoolType	

== RegkeyEffectiveRights53Behaviors ==

The RegkeyEffectiveRights53Behaviors complex type defines a number of behaviors that allow a more detailed definition of the registrykeyeffectiverights53_object being specified. 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.

The RegkeyEffectiveRights53Behaviors extend the win-def:RegistryBehaviors and therefore include the behaviors defined by that type.

Extends: win-def:RegistryBehaviors

Attributes

Table 680: Attributes

Attribute	Type	Desc.
include_group (Deprecated)	xsd:boolean (optional *default*='true')	'include_group' defines whether the group SID should be included in the object when the object is defined by a group SID. For example, the intent of an object defined by a group SID might be to retrieve all the user SIDs that are a member of the group, but not the group SID itself.
resolve_group (Deprecated)	xsd:boolean (optional *default*='false')	The 'resolve_group' behavior defines whether an object set defined by a group SID should be resolved to return a set that contains all the user SIDs that are a member of that group. Note that all child groups should also be resolved any valid domain users that are members of the group should also be included. The intent of this behavior is to end up with a list of all individual users from that system that make up the group once everything has been resolved.

< regkeyeffectiverights_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.3
- Reason: Replaced by the regkeyeffectiverights53_test. This test uses a trustee_name element for identifying trustees. Trustee names are not unique, and a new test was created to use trustee SIDs, which are unique. See the regkeyeffectiverights53_test.
- Comment: This test has been deprecated and will be removed in version 6.0 of the language.

The registry key effective rights test is used to check the effective rights associated with Windows files. Note that the trustee's effective access rights are the access rights that the DACL grants to the trustee or to any groups of which the trustee is a member. The regkeyeffectiverights_test element extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a regkeyeffectiverights_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 681: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< regkeyeffectiverights_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.3
- Reason: Replaced by the regkeyeffectiverights53_object. This object uses a trustee_name element for identifying trustees. Trustee names are not unique, and a new object was created to use trustee SIDs, which are unique. See the regkeyeffectiverights53_object.
- Comment: This object has been deprecated and will be removed in version 6.0 of the language.

Extends: oval-def:ObjectType

Child Elements

Table 682: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	win-def:RegkeyEffectiveRightsBehaviors (0..1)	
hive	win-def:EntityObjectRegistryHiveType (1..1)	The hive that the registry key belongs to. This is restricted to a specific set of values: HKEY_CLASSES_ROOT, HKEY_CURRENT_CONFIG, HKEY_CURRENT_USER, HKEY_CURRENT_USER_LOCAL_SETTINGS,HKEY_LOCAL_MACHINE, and HKEY_USERS.
key	oval-def:EntityObjectStringType (1..1)	The key element describes a registry key to be collected. Note that the hive portion of the string should not be included, as this data should be found under the hive element.
trustee_name	oval-def:EntityObjectStringType (1..1)	The trustee_name element is the unique name that associated a particular SID. A SID can be associated with a user, group, or program (such as a Windows service). In Windows, trustee names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, trustee names should be identified in the form: “domaintrustee name”. For local trustee names use: “computer nametrustee name”. For built-in accounts on the system, use the trustee name without a domain.

< regkeyeffectiverights_state > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.3
- Reason: Replaced by the regkeyeffectiverights53_state. This state uses a trustee_name element for identifying trustees. Trustee names are not unique, and a new state was created to use trustee SIDs, which are unique. See the regkeyeffectiverights53_state.
- Comment: This state has been deprecated and will be removed in version 6.0 of the language.

The regkeyeffectiverights_state element defines the different rights that can be associated with a given regkeyeffectiverights_object. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 683: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
hive	win-def:EntityStateRegistryHiveType (0..1)	This element specifies the hive of a registry key on the machine from which to retrieve the SACL.
key	oval-def:EntityStateStringType (0..1)	This element specifies a registry key on the machine from which to retrieve the SACL. Note that the hive portion of the string should not be inclueded, as this data should be found under the hive element.
trustee_name	oval-def:EntityStateStringType (0..1)	The unique name associated with a particular security identifier (SID). In Windows, trustee names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, trustee names should be identified in the form: “domaintrustee name”. For local trustee names use: “computer nametrustee name”. For built-in accounts on the system, use the trustee name without a domain.
standard_delete	oval-def:EntityStateBoolType (0..1)	The right to delete the object.
standard_read_control	oval-def:EntityStateBoolType (0..1)	The right to read the information in the object’s Security Descriptor, not including the information in the SACL.
standard_write_dac	oval-def:EntityStateBoolType (0..1)	The right to modify the DACL in the object’s Security Descriptor.
standard_write_owner	oval-def:EntityStateBoolType (0..1)	The right to change the owner in the object’s Security Descriptor.
standard_synchronize	oval-def:EntityStateBoolType (0..1)	The right to use the object for synchronization. This enables a thread to wait until the object is in the signaled state. Some object types do not support this access right.
access_system_security	oval-def:EntityStateBoolType (0..1)	Indicates access to a system access control list (SACL).
generic_read	oval-def:EntityStateBoolType (0..1)	Read access.
generic_write	oval-def:EntityStateBoolType (0..1)	Write access.
generic_execute	oval-def:EntityStateBoolType (0..1)	Execute access.
generic_all	oval-def:EntityStateBoolType (0..1)	Read, write, and execute access.
key_query_value	oval-def:EntityStateBoolType (0..1)	
key_set_value	oval-def:EntityStateBoolType (0..1)	
key_create_sub_key	oval-def:EntityStateBoolType (0..1)	
key_enumerate_sub_keys	oval-def:EntityStateBoolType (0..1)	

== RegkeyEffectiveRightsBehaviors == (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.3
- Reason: Replaced by the RegkeyEffectiveRightsBehaviors53. The RegkeyEffectiveRightsBehaviors complex type is used by the regkeyeffectiverights_test which uses a trustee_name element for identifying trustees. Trustee names are not unique, and a new test was created to use trustee SIDs, which are unique. This new test utilizes the RegkeyEffectiveRightsBehaviors53 complex type, and as a result, the RegkeyEffectiveRightsBehaviors complex type is no longer needed.
- Comment: This complex type has been deprecated and will be removed in version 6.0 of the language.

The RegkeyEffectiveRightsBehaviors complex type defines a number of behaviors that allow a more detailed definition of the registrykeyeffectiverights_object being specified. 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.

The RegkeyEffectiveRightsBehaviors extend the win-def:RegistryBehaviors and therefore include the behaviors defined by that type.

Extends: win-def:RegistryBehaviors

Attributes

Table 684: Attributes

Attribute	Type	Desc.
include_group (Deprecated)	xsd:boolean (optional *default*='true')	'include_group' defines whether the group trustee name should be included in the object when the object is defined by a group trustee name. For example, the intent of an object defined by a group trustee name might be to retrieve all the user trustee names that are members of the group, but not the group trustee name itself.
resolve_group (Deprecated)	xsd:boolean (optional *default*='false')	The 'resolve_group' behavior defines whether an object set defined by a group SID should be resolved to return a set that contains all the user SIDs that are a member of that group. Note that all child groups should also be resolved any valid domain users that are members of the group should also be included. The intent of this behavior is to end up with a list of all individual users from that system that make up the group once everything has been resolved.

< service_test >

The service_test is used to check metadata associated with Windows services. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a service_object and the optional state elements specify the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 685: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< service_object >

The service_object element is used by a service_test to define the specific service(s) to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 686: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
service_name	oval-def:EntityObjectStringType (1..1)	The service_name element specifies the service name as stored in the Service Control Manager (SCM) database on the system.
oval-def:filter	n/a (0..unbounded)	

< service_state >

The service_state element defines the different metadata associated with a Windows service. This includes the service name, display name, description, type, start type, current state, controls accepted, start name, path, pid, service flag, and dependencies. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 687: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
service_name	oval-def:EntityStateStringType (0..1)	The service_name element specifies the name of the service as specified in the Service Control Manager (SCM) database.
display_name	oval-def:EntityStateStringType (0..1)	The display_name element specifies the name of the service as specified in tools such as Control Panel->Administrative Tools->Services.
description	oval-def:EntityStateStringType (0..1)	The description element specifies the description of the service.
service_type	win-def:EntityStateServiceTypeType (0..1)	The service_type element specifies the type of the service.
start_type	win-def:EntityStateServiceStartTypeType (0..1)	The start_type element specifies when the service should be started.
current_state	win-def:EntityStateServiceCurrentStateType (0..1)	The current_state element specifies the current state of the service.
controls_accepted	win-def:EntityStateServiceControlsAcceptedType (0..1)	The controls_accepted element specifies the control codes that a service will accept and process.
start_name	oval-def:EntityStateStringType (0..1)	The start_name element specifies the account under which the process should run.
path	oval-def:EntityStateStringType (0..1)	The path element specifies the path to the binary of the service.
pid	oval-def:EntityStateIntType (0..1)	The pid element specifies the process ID of the service.
service_flag	oval-def:EntityStateBoolType (0..1)	The service_flag element specifies if the service is in a system process that must always run (1) or if the service is in a non-system process or is not running (0). If the service is not running, the pid will be 0. Otherwise, the pid will be non-zero.
dependencies	oval-def:EntityStateStringType (0..1)	The dependencies element specifies the dependencies of this service on other services.

< serviceeffectiverights_test >

The service effective rights test is used to check the effective rights associated with Windows services. Note that the trustee's effective access rights are the access rights that the DACL grants to the trustee or to any groups of which the trustee is a member. The serviceeffectiverights_test element extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a serviceeffectiverights_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 688: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< serviceeffectiverights_object >

The serviceeffectiverights_object element is used by the serviceeffectiverights_test to define the objects used to evaluate against the specified state. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic.

A serviceeffectiverights_object is defined as a combination of a Windows service_name and trustee_sid. The service_name entity represents the service to be evaluated while the trustee_sid entity represents the account (SID) to check the effective rights of. If multiple services or SIDs are matched by either reference, then each possible combination of service and SID is a matching service effective rights object.

Extends: oval-def:ObjectType

Child Elements

Table 689: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	win-def:ServiceEffectiveRightsBehaviors (0..1)	
service_name	oval-def:EntityObjectStringType (1..1)	The service_name element describes a service to be collected. Note that the service_name element should contain the actual name of the service and not its display name that is found in Control Panel->Administrative Tools->Services. For example, if you wanted to check the effective rights of the Automatic Updates service you would specify 'wuauserv' for the service_name element not 'Automatic Updates'.
trustee_sid	oval-def:EntityObjectStringType (1..1)	The trustee_sid entity identifies a set of SIDs associated with a user, group, system, or program (such as a Windows service). If an operation other than equals is used to identify matching trustees (i.e. not equal, or a pattern match) then the resulting matches shall be limited to only the trustees referenced in the service's Security Descriptor. The scope is limited here to avoid unnecessarily resource intensive searches for trustees. Note that the larger scope of all known trustees may be obtained through the use of variables.
oval-def:filter	n/a (0..unbounded)	

< serviceeffectiverights_state >

The serviceeffectiverights_state element defines the different rights that can be associated with a given serviceeffectiverights_object. Please refer to the individual elements in the schema for more details about what each represents.

See http://support.microsoft.com/kb/914392 for more information.

Extends: oval-def:StateType

Child Elements

Table 690: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
service_name	oval-def:EntityStateStringType (0..1)	The service_name element specifies a service on the machine from which to retrieve the DACL. Note that the service_name element should contain the actual name of the service and not its display name that is found in Control Panel->Administrative Tools->Services. For example, if you wanted to check the effective rights of the Automatic Updates service you would specify 'wuauserv' for the service_name element not 'Automatic Updates'.
trustee_sid	oval-def:EntityStateStringType (0..1)	The trustee_sid element is the unique SID that is associated with a user, group, system, or program (such as a Windows service).
standard_delete	oval-def:EntityStateBoolType (0..1)	This permission is required to call the DeleteService function to delete the service.
standard_read_control	oval-def:EntityStateBoolType (0..1)	This permission is required to call the QueryServiceObjectSecurity function to query the Security Descriptor of the service object.
standard_write_dac	oval-def:EntityStateBoolType (0..1)	This permission is required to call the SetServiceObjectSecurity function to modify the DACL member of the service object's Security Descriptor.
standard_write_owner	oval-def:EntityStateBoolType (0..1)	This permission is required to call the SetServiceObjectSecurity function to modify the Owner and Group members of the service object's Security Descriptor.
generic_read	oval-def:EntityStateBoolType (0..1)	Read access (STANDARD_RIGHTS_READ, SERVICE_QUERY_CONFIG, SERVICE_QUERY_STATUS, SERVICE_INTERROGATE, SERVICE_ENUMERATE_DEPENDENTS).
generic_write	oval-def:EntityStateBoolType (0..1)	Write access (STANDARD_RIGHTS_WRITE, SERVICE_CHANGE_CONFIG).
generic_execute	oval-def:EntityStateBoolType (0..1)	Execute access (STANDARD_RIGHTS_EXECUTE, SERVICE_START, SERVICE_STOP, SERVICE_PAUSE_CONTINUE, SERVICE_USER_DEFINED_CONTROL).
service_query_conf	oval-def:EntityStateBoolType (0..1)	This permission is required to call the QueryServiceConfig and QueryServiceConfig2 functions to query the service configuration.
service_change_conf	oval-def:EntityStateBoolType (0..1)	This permission is required to call the ChangeServiceConfig or ChangeServiceConfig2 function to change the service configuration.
service_query_stat	oval-def:EntityStateBoolType (0..1)	This permission is required to call the QueryServiceStatusEx function to ask the service control manager about the status of the service.
service_enum_dependents	oval-def:EntityStateBoolType (0..1)	This permission is required to call the EnumDependentServices function to enumerate all the services dependent on the service.
service_start	oval-def:EntityStateBoolType (0..1)	This permission is required to call the StartService function to start the service.
service_stop	oval-def:EntityStateBoolType (0..1)	This permission is required to call the ControlService function to stop the service.
service_pause	oval-def:EntityStateBoolType (0..1)	This permission is required to call the ControlService function to pause or continue the service.
service_interrogate	oval-def:EntityStateBoolType (0..1)	This permission is required to call the ControlService function to ask the service to report its status immediately.

== ServiceEffectiveRightsBehaviors ==

The ServiceEffectiveRightsBehaviors complex type defines a number of behaviors that allow a more detailed definition of the serviceeffectiverights_object being specified. 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.

Attributes

Table 691: Attributes

Attribute	Type	Desc.
include_group (Deprecated)	xsd:boolean (optional *default*='true')	'include_group' defines whether the group trustee sid should be included in the object when the object is defined by a group trustee sid. For example, the intent of an object defined by a group trustee sid might be to retrieve all the user trustee sids that are members of the group, but not the group trustee sid itself.
resolve_group (Deprecated)	xsd:boolean (optional *default*='false')	The 'resolve_group' behavior defines whether an object set defined by a group SID should be resolved to return a set that contains all the user SIDs that are a member of that group. Note that all child groups should also be resolved any valid domain users that are members of the group should also be included. The intent of this behavior is to end up with a list of all individual users from that system that make up the group once everything has been resolved.

< sharedresource_test >

The shared resource test is used to check properties associated with any shared resource on the system. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a sharedresource_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 692: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< sharedresource_object >

The sharedresource_object element is used by a shared resource test to define the object, in this case a shared resource, to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An shared resource object consists of a single netname entity that identifies a specific shared resource.

Extends: oval-def:ObjectType

Child Elements

Table 693: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
netname	oval-def:EntityObjectStringType (1..1)	The netname element is the unique name that is associated with a specific shared resource.
oval-def:filter	n/a (0..unbounded)	

< sharedresource_state >

The sharedresource_state element defines the different metadata associated with a Windows shared resource. This includes the share type, permissions, and max uses. This state mirrors the SHARE_INFO_2 structure. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 694: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
netname	oval-def:EntityStateStringType (0..1)	This element specifies the name associated with a particular shared resource.
shared_type	win-def:EntityStateSharedResourceTypeType (0..1)	The type of the shared resource.
max_uses	oval-def:EntityStateIntType (0..1)	The maximum number of concurrent connections that the shared resource can accommodate.
current_uses	oval-def:EntityStateIntType (0..1)	The number of current connections to the resource.
local_path	oval-def:EntityStateStringType (0..1)	The local path for the shared resource.
access_read_permission	oval-def:EntityStateBoolType (0..1)	Permission to read data from a resource and, by default, to execute the resource.
access_write_permission	oval-def:EntityStateBoolType (0..1)	Permission to write data to the resource.
access_create_permission	oval-def:EntityStateBoolType (0..1)	Permission to create an instance of the resource (such as a file); data can be written to the resource as the resource is created.
access_exec_permission	oval-def:EntityStateBoolType (0..1)	Permission to execute the resource.
access_delete_permission	oval-def:EntityStateBoolType (0..1)	Permission to delete the resource.
access_atrib_permission	oval-def:EntityStateBoolType (0..1)	Permission to modify the resource's attributes (such as the date and time when a file was last modified).
access_perm_permission	oval-def:EntityStateBoolType (0..1)	Permission to modify the permissions (read, write, create, execute, and delete) assigned to a resource for a user or application.
access_all_permission	oval-def:EntityStateBoolType (0..1)	Permission to read, write, create, execute, and delete resources, and to modify their attributes and permissions.

< sharedresourceauditedpermissions_test >

The shared resource audited permissions test is used to check the audit permissions associated with any shared resource on the system. Note that the trustee's audited permissions are the audit permissons that the SACL grants to the trustee or to any groups of which the trustee is a member. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a sharedresourceauditedpermissions_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 695: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< sharedresourceauditedpermissions_object >

The sharedresourceauditedpermissions_object element is used by a shared resource audited permissions test to define the objects used to evaluate against the specified state. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic.

A shared resource audited permissions object consists of a netname entity that identifies a specific shared resource and a trustee_sid entity that identifies a specific account (SID) to check the audited permissions of.

Extends: oval-def:ObjectType

Child Elements

Table 696: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	win-def:SharedResourceAuditedPermissionsBehaviors (0..1)	
netname	oval-def:EntityObjectStringType (1..1)	The netname element is the unique name that is associated with a specific shared resource.
trustee_sid	oval-def:EntityObjectStringType (1..1)	The trustee_sid entity identifies a unique SID associated with a user, group, system, or program (such as a Windows service). If an operation other than equals is used to identify matching trustees (i.e. not equal, or a pattern match) then the resulting matches shall be limited to only the trustees referenced in the file's Security Descriptor. The scope is limited here to avoid unnecessarily resource intensive searches for trustees. Note that the larger scope of all known trustees may be obtained through the use of variables.
oval-def:filter	n/a (0..unbounded)	

< sharedresourceauditedpermissions_state >

The sharedresourceauditedpermissions_state element defines the different audited permissions that can be associated with a given sharedresourceauditedpermissions_object. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 697: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
netname	oval-def:EntityStateStringType (0..1)	This element specifies the name associated with a particular shared resource.
trustee_sid	oval-def:EntityStateStringType (0..1)	The trustee_sid element is the unique SID that associated a user, group, system, or program (such as a Windows service).
standard_delete	win-def:EntityStateAuditType (0..1)	The right to delete the object.
standard_read_control	win-def:EntityStateAuditType (0..1)	The right to read the information in the object's Security Descriptor, not including the information in the SACL.
standard_write_dac	win-def:EntityStateAuditType (0..1)	The right to modify the DACL in the object's Security Descriptor.
standard_write_owner	win-def:EntityStateAuditType (0..1)	The right to change the owner in the object's Security Descriptor.
standard_synchronize	win-def:EntityStateAuditType (0..1)	The right to use the object for synchronization. This enables a thread to wait until the object is in the signaled state. Some object types do not support this access right.
access_system_security	win-def:EntityStateAuditType (0..1)	Indicates access to a system access control list (SACL).
generic_read	win-def:EntityStateAuditType (0..1)	Read access.
generic_write	win-def:EntityStateAuditType (0..1)	Write access.
generic_execute	win-def:EntityStateAuditType (0..1)	Execute access.
generic_all	win-def:EntityStateAuditType (0..1)	Read, write, and execute access.

== SharedResourceAuditedPermissionsBehaviors ==

The SharedResourceAuditedPermissionsBehaviors complex type defines a behavior that allows for a more detailed definition of the sharedresourceauditedpermissions_object being specified. Note that using this behavior 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.

Attributes

Table 698: Attributes

Attribute	Type	Desc.
include_group (Deprecated)	xsd:boolean (optional *default*='true')	'include_group' defines whether the group SID should be included in the object when the object is defined by a group SID. For example, the intent of an object defined by a group SID might be to retrieve all the user SIDs that are a member of the group, but not the group SID itself.

< sharedresourceeffectiverights_test >

The shared resource effective rights test is used to check the effective rights associated with any shared resource on the system. Note that the trustee's effective access rights are the access rights that the DACL grants to the trustee or to any groups of which the trustee is a member. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a sharedresourceeffectiverights_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 699: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< sharedresourceeffectiverights_object >

The sharedresourceeffectiverights_object element is used by a shared resource effective rights test to define the object, in this case a shared resource effective rights object, to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A shared resource effective rights object consists of a netname entity that identifies a specific shared resource and a trustee_sid entity that identifies a specific account (SID) to check the effective rights of.

Extends: oval-def:ObjectType

Child Elements

Table 700: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	win-def:SharedResourceEffectiveRightsBehaviors (0..1)	
netname	oval-def:EntityObjectStringType (1..1)	The netname element is the unique name that is associated with a specific shared resource.
trustee_sid	oval-def:EntityObjectStringType (1..1)	The trustee_sid entity identifies a unique SID associated with a user, group, system, or program (such as a Windows service). If an operation other than equals is used to identify matching trustees (i.e. not equal, or a pattern match) then the resulting matches shall be limited to only the trustees referenced in the file's Security Descriptor. The scope is limited here to avoid unnecessarily resource intensive searches for trustees. Note that the larger scope of all known trustees may be obtained through the use of variables.
oval-def:filter	n/a (0..unbounded)	

< sharedresourceeffectiverights_state >

The sharedresourceeffectiverights_state element defines the different rights that can be associated with a given sharedresourceeffectiverights_object. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 701: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
netname	oval-def:EntityStateStringType (0..1)	This element specifies the name associated with a particular shared resource.
trustee_sid	oval-def:EntityStateStringType (0..1)	The trustee_sid element is the unique SID that associated a user, group, system, or program (such as a Windows service).
standard_delete	oval-def:EntityStateBoolType (0..1)	The right to delete the object.
standard_read_control	oval-def:EntityStateBoolType (0..1)	The right to read the information in the object's Security Descriptor, not including the information in the SACL.
standard_write_dac	oval-def:EntityStateBoolType (0..1)	The right to modify the DACL in the object's Security Descriptor.
standard_write_owner	oval-def:EntityStateBoolType (0..1)	The right to change the owner in the object's Security Descriptor.
standard_synchronize	oval-def:EntityStateBoolType (0..1)	The right to use the object for synchronization. This enables a thread to wait until the object is in the signaled state. Some object types do not support this access right.
access_system_security	oval-def:EntityStateBoolType (0..1)	Indicates access to a system access control list (SACL).
generic_read	oval-def:EntityStateBoolType (0..1)	Read access.
generic_write	oval-def:EntityStateBoolType (0..1)	Write access.
generic_execute	oval-def:EntityStateBoolType (0..1)	Execute access.
generic_all	oval-def:EntityStateBoolType (0..1)	Read, write, and execute access.

== SharedResourceEffectiveRightsBehaviors ==

The SharedResourceEffectiveRightsBehaviors complex type defines a behavior that allows for a more detailed definition of the sharedresourceeffectiverights_object being specified. Note that using this behavior 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.

Attributes

Table 702: Attributes

Attribute	Type	Desc.
include_group (Deprecated)	xsd:boolean (optional *default*='true')	'include_group' defines whether the group SID should be included in the object when the object is defined by a group SID. For example, the intent of an object defined by a group SID might be to retrieve all the user SIDs that are a member of the group, but not the group SID itself.

< sid_test >

The SID test is used to check properties associated with the specified SID. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a sid_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 703: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< sid_object >

The sid_object element is used by a sid_test to define the object set, in this case a set of SIDs (identified by name), to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 704: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	win-def:SidBehaviors (0..1)	
trustee_name	oval-def:EntityObjectStringType (1..1)	The trustee_name element is the unique name that associated a particular SID. A SID can be associated with a user, group, or program (such as a Windows service). In Windows, trustee names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, trustee names should be identified in the form: “domaintrustee name”. For local trustee names use: “computer nametrustee name”. For built-in accounts on the system, use the trustee name without a domain.
oval-def:filter	n/a (0..unbounded)	

< sid_state >

The sid_state element defines the different metadata associate with a Windows trustee (identified by name). Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 705: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
trustee_name	oval-def:EntityStateStringType (0..1)	This element specifies the trustee name associated with a particular SID. In Windows, trustee names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, trustee names should be identified in the form: “domaintrustee name”. For local trustee names use: “computer nametrustee name”. For built-in accounts on the system, use the trustee name without a domain.
trustee_sid	oval-def:EntityStateStringType (0..1)	The security identifier (SID) of the specified trustee name.
trustee_domain	oval-def:EntityStateStringType (0..1)	The domain of the specified trustee name.

== SidBehaviors ==

The SidBehaviors complex type defines a number of behaviors that allow a more detailed definition of the sid_object being specified. 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.

Attributes

Table 706: Attributes

Attribute	Type	Desc.
include_group	xsd:boolean (optional *default*='true')	'include_group' defines whether the group SID should be included in the object when the object is defined by a group SID. For example, the intent of an object defined by a group SID might be to retrieve all the user SIDs that are a member of the group, but not the group SID itself.
resolve_group	xsd:boolean (optional *default*='false')	The 'resolve_group' behavior defines whether an object set defined by a group SID should be resolved to return a set that contains all the user SIDs that are a member of that group. Note that all child groups should also be resolved any valid domain users that are members of the group should also be included. The intent of this behavior is to end up with a list of all individual users from that system that make up the group once everything has been resolved.

< sid_sid_test >

The sid_sid_test is used to check properties associated with the specified SID. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a sid_sid_object and the optional state element specifies the metadata to check.

Note that this sid_sid test was added in version 5.4 as a temporary fix. There is a need within the community to identify things like users and groups by both the name and the SID. For version 6 of OVAL, work is underway for a better solution to the problem, but for now, a second test was added to satisfy the need.

Extends: oval-def:TestType

Child Elements

Table 707: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< sid_sid_object >

The sid_sid_object element is used by a sid_sid_test to define the object set, in this case a set of SIDs, to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the

ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 708: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	win-def:SidSidBehaviors (0..1)	
trustee_sid	oval-def:EntityObjectStringType (1..1)	The trustee_sid entity identifies a unique SID associated with a user, group, system, or program (such as a Windows service).
oval-def:filter	n/a (0..unbounded)	

< sid_sid_state >

The sid_state element defines the different metadata associate with a Windows trustee (identified by SID). Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 709: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
trustee_sid	oval-def:EntityStateStringType (0..1)	The security identifier (SID) of the specified trustee name.
trustee_name	oval-def:EntityStateStringType (0..1)	This element specifies the trustee name associated with a particular SID. In Windows, trustee names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, trustee names should be identified in the form: “domaintrustee name”. For local trustee names use: “computer nametrustee name”. For built-in accounts on the system, use the trustee name without a domain.
trustee_domain	oval-def:EntityStateStringType (0..1)	The domain of the specified trustee name.

== SidSidBehaviors ==

The SidSidBehaviors complex type defines a number of behaviors that allow a more detailed definition of the sid_sid_object being specified. 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.

Attributes

Table 710: Attributes

Attribute	Type	Desc.
include_group	xsd:boolean (optional *default*='true')	'include_group' defines whether the group SID should be included in the object when the object is defined by a group SID. For example, the intent of an object defined by a group SID might be to retrieve all the user SIDs that are a member of the group, but not the group SID itself.
resolve_group	xsd:boolean (optional *default*='false')	The 'resolve_group' behavior defines whether an object set defined by a group SID should be resolved to return a set that contains all the user SIDs that are a member of that group. Note that all child groups should also be resolved any valid domain users that are members of the group should also be included. The intent of this behavior is to end up with a list of all individual users from that system that make up the group once everything has been resolved.

< systemmetric_test >

The system metric test is used to check the value of a particular Windows system metric. Access to this information is exposed by the GetSystemMetrics function in User32.dll.

Extends: oval-def:TestType

Child Elements

Table 711: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< systemmetric_object >

The system metric object element is used by a system metric test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 712: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
index	win-def:EntityObjectSystemMetricIndexType (1..1)	The index entity provides the system metric index value that is desired.
oval-def:filter	n/a (0..unbounded)	

< systemmetric_state >

The system metric state element defines the different information that can be found in a Windows system metric value. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 713: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
index	win-def:EntityStateSystemMetricIndexType (0..1)	The index entity corresponds to the systemmetric_object index entity.
value	oval-def:EntityStateIntType (0..1)	The optional value entity provides the value of the system metric that is expected.

< uac_test >

The user access control test is used to check setting related to User Access Control within Windows. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a uaac_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 714: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< uac_object >

The uac_object element is used by a user access control test to define those objects to evaluate based on a specified state. There is actually only one object relating to user access control and this is the system as a whole. Therefore, there are no child entities defined. Any OVAL Test written to check user access control settings will reference the same uac_object which is basically an empty object element.

Extends: oval-def:ObjectType

< uac_state >

The uac_state element specifies the different settings that are available under User Access Control. A user access control test will reference a specific instance of this state that defines the exact settings that need to be evaluated. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 715: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
admin_approval_mode	oval-def:EntityStateBoolType (0..1)	Admin Approval Mode for the Built-in Administrator account.
elevation_prompt_admin	oval-def:EntityStateStringType (0..1)	Behavior of the elevation prompt for administrators in Admin Approval Mode.
elevation_prompt_standard	oval-def:EntityStateStringType (0..1)	Behavior of the elevation prompt for standard users.
detect_installations	oval-def:EntityStateBoolType (0..1)	Detect application installations and prompt for elevation.
elevate_signed_executables	oval-def:EntityStateBoolType (0..1)	Only elevate executables that are signed and validated.
elevate_uiaccess	oval-def:EntityStateBoolType (0..1)	Only elevate UIAccess applications that are installed in secure locations.
run_admins_aam	oval-def:EntityStateBoolType (0..1)	Run all administrators in Admin Approval Mode.
secure_desktop	oval-def:EntityStateBoolType (0..1)	Switch to the secure desktop when prompting for elevation.
virtualize_write_failures	oval-def:EntityStateBoolType (0..1)	Virtualize file and registry write failures to per-user locations.

< user_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.11
- Reason: Replaced by the user_sid55_test. This test uses trustee names for identifying accounts on the system. Trustee names are not unique and the user_sid55_test, which uses trustee SIDs which are unique, should be used instead. See the user_sid55_test.
- Comment: This test has been deprecated and will be removed in version 6.0 of the language.

The user_test is used to check information about Windows users. When the user_test collects the users on the system, it should only include the local and built-in user accounts and not domain user accounts. However, it is important to note that domain user accounts can still be looked up. Also, note that the collection of groups, for which a user is a member, is not recursive. The only groups that will be collected are those for which the user is a direct member. For example, if a user is a member of group A, and group A is a member of group B, the only group that will be collected is group A. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a user_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 716: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< user_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.11
- Reason: Replaced by the user_sid55_object. This object uses trustee names for identifying accounts on the system. Trustee names are not unique and the user_sid55_object, which uses trustee SIDs which are unique, should be used instead. See the user_sid55_object.
- Comment: This object has been deprecated and will be removed in version 6.0 of the language.

Extends: oval-def:ObjectType

Child Elements

Table 717: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
user	oval-def:EntityObjectStringType (1..1)	The user entity holds a string that represents the name of a particular user. In Windows, user names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, users should be identified in the form: "domainuser name". For local users use: "computer nameuser name". For built-in accounts on the system, use the user name without a domain.
oval-def:filter	n/a (0..unbounded)	

< user_state > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.11
- Reason: Replaced by the user_sid55_state. This state uses trustee names for identifying accounts on the system. Trustee names are not unique and the user_sid55_state, which uses trustee SIDs which are unique, should be used instead. See the user_sid55_state.
- Comment: This state has been deprecated and will be removed in version 6.0 of the language.

The user_state element enumerates the different groups (identified by name) that a Windows user might belong to. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 718: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
user	oval-def:EntityStateStringType (0..1)	The user entity holds a string that represents the name of a particular user. In Windows, user names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, users should be identified in the form: "domainuser name". For local users use: "computer nameuser name". For built-in accounts on the system, use the user name without a domain.
enabled	oval-def:EntityStateBoolType (0..1)	This element holds a boolean value that specifies whether the particular user account is enabled or not.
group	oval-def:EntityStateStringType (0..1)	A string that represents the name of a particular group. In Windows, group names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, groups should be identified in the form: "domaingroup name". For local groups use: "computer namegroup name". For built-in accounts on the system, use the group name without a domain.The group element can be included multiple times in a system characteristic item in order to record that a user can be a member of a number of different groups. Note that the entity_check attribute associated with EntityStateStringType guides the evaluation of entities like group that refer to items that can occur an unbounded number of times.
last_logon	oval-def:EntityStateIntType (0..1)	The date and time when the last logon occurred. This value is stored as the number of seconds that have elapsed since 00:00:00, January 1, 1970, GMT. If the target system is a domain controller, this data is maintained separately on each backup domain controller (BDC) in the domain. To obtain an accurate value, you must query each BDC in the domain. The last logoff occurred at the time indicated by the largest retrieved value.
full_name	oval-def:EntityStateStringType (0..1)	A Unicode string that contains the full name of the user. This string can be a NULL string, or it can have any number of characters before the terminating null character.
comment	oval-def:EntityStateStringType (0..1)	A Unicode string that contains a comment to associate with the user account. The string can be a NULL string, or it can have any number of characters before the terminating null character.
password_age_days	oval-def:EntityStateIntType (0..1)	The number of days that have elapsed since the password was last changed. This data should be rounded up to the nearest integer.
lockout	oval-def:EntityStateBoolType (0..1)	The account is currently locked out.
passwd_notreqd	oval-def:EntityStateBoolType (0..1)	No password is required.
dont_expire_passwd	oval-def:EntityStateBoolType (0..1)	The password should never expire on the account.
encrypted_text_password_allowed	oval-def:EntityStateBoolType (0..1)	The user's password is stored under reversible encryption in the Active Directory.
not_delegated	oval-def:EntityStateBoolType (0..1)	Marks the account as "sensitive"; other users cannot act as delegates of this user account.
use_des_key_only	oval-def:EntityStateBoolType (0..1)	Restrict this principal to use only Data Encryption Standard (DES) encryption types for keys.
dont_require_preauth	oval-def:EntityStateBoolType (0..1)	This account does not require Kerberos preauthentication for logon.
pass-	oval-	The password expiration information. Zero if the password has not expired (and nonzero if it has).

< user_sid55_test >

The user_sid55_test is used to check information about Windows users. When the user_sid55_test collects the user SIDs on the system, it should only include the local and built-in user SIDs and not domain user SIDs. However, it is important to note that domain user SIDs can still be looked up. Also, note that the collection of groups, for which a user is a member, is not recursive. The only groups that will be collected are those for which the user is a direct member. For example, if a user is a member of group A, and group A is a member of group B, the only group that will be collected is group A. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a user_sid55_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 719: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< user_sid55_object >

The user_sid55_object represents a set of users on a Windows system. This set (which might contain only one user) is identified by a SID.

Extends: oval-def:ObjectType

Child Elements

Table 720: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
user_sid	oval-def:EntityObjectStringType (1..1)	The user_sid entity holds a string that represents the SID of a particular user.
oval-def:filter	n/a (0..unbounded)	

< user_sid55_state >

The user_sid55_state element enumerates the different groups (identified by SID) that a Windows user might belong to. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 721: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
user_sid	oval-def:EntityStateStringType (0..1)	The user_sid entity holds a string that represents the SID of a particular user.
enabled	oval-def:EntityStateBoolType (0..1)	This element holds a boolean value that specifies whether the particular user account is enabled or not.
group_sid	oval-def:EntityStateStringType (0..1)	A string the represents the SID of a particular group. The group_sid element can be included multiple times in a system characteristic item in order to record that a user can be a member of a number of different groups. Note that the entity_check attribute associated with EntityStateStringType guides the evaluation of entities like group that refer to items that can occur an unbounded number of times.
last_logon	oval-def:EntityStateIntType (0..1)	The date and time when the last logon occurred. This value is stored as the number of seconds that have elapsed since 00:00:00, January 1, 1970, GMT.

< user_sid_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.5
- Reason: Replaced by the user_sid55_test. This test uses user and group elements that are incorrectly named. A new test was created to change the element names to their correct values which are user_sid and group_sid. See the user_sid55_test.
- Comment: This test has been deprecated and will be removed in version 6.0 of the language.

The user_sid_test is used to check information about Windows users. When the user_sid_test collects the user SIDs on the system, it should only include the local and built-in user SIDs and not domain user SIDs. However, it is important to note that domain user SIDs can still be looked up. Also, note that the collection of groups, for which a user is a member, is not recursive. The only groups that will be collected are those for which the user is a direct member. For example, if a user is a member of group A, and group A is a member of group B, the only group that will be collected is group A. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a user_sid_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 722: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< user_sid_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.5
- Reason: Replaced by the user_sid55_object. This object uses a user element that is incorrectly named. A new object was created to change the element name to its correct value which is user_sid. See the user_sid55_object.
- Comment: This object has been deprecated and will be removed in version 6.0 of the language.

The user_sid_object represents a set of users on a Windows system. This set (which might contain only one user) is identified by a SID.

Extends: oval-def:ObjectType

Child Elements

Table 723: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
user	oval-def:EntityObjectStringType (1..1)	The user_sid entity holds a string that represents the SID of a particular user.

< user_sid_state > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.5
- Reason: Replaced by the user_sid55_state. This state uses user and group elements that are incorrectly named. A new state was created to change the element names to their correct values which are user_sid and group_sid. See the user_sid55_state.
- Comment: This state has been deprecated and will be removed in version 6.0 of the language.

The user_sid_state element enumerates the different groups (identified by SID) that a Windows user might belong to. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 724: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
user	oval-def:EntityStateStringType (0..1)	The user_sid entity holds a string that represents the SID of a particular user.
enabled	oval-def:EntityStateBoolType (0..1)	This element holds a boolean value that specifies whether the particular user account is enabled or not.
group	oval-def:EntityStateStringType (0..1)	A string the represents the SID of a particular group. The group_sid element can be included multiple times in a system characteristic item in order to record that a user can be a member of a number of different groups. Note that the entity_check attribute associated with EntityStateStringType guides the evaluation of entities like group that refer to items that can occur an unbounded number of times.

< userright_test >

The userright_test is used to enumerate all of the trustees/SIDs that have been granted a specific user right/privilege.

Extends: oval-def:TestType

Child Elements

Table 725: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< userright_object >

The userright_object is used to collect the trustees/SIDs that have been granted a specific user right/privilege.

Extends: oval-def:ObjectType

Child Elements

Table 726: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
userright	win-def:EntityObjectUserRightType (1..1)	The userright entity holds a string that represents the name of a particular user right/privilege.
oval-def:filter	n/a (0..unbounded)	

< userright_state >

Extends: oval-def:StateType

Child Elements

Table 727: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
userright	win-def:EntityStateUserRightType (0..1)	The userright entity holds a string that represents the name of a particular user right/privilege.
trustee_name	oval-def:EntityStateStringType (0..1)	The trustee_name entity is the unique name associated with the SID that has been granted the specified user right/privilege. A trustee can be associated with a user, group, or program (such as a Windows service). In Windows, trustee names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, trustee names should be identified in the form: “domaintrustee name”. For local trustee names use: “computer nametrustee name”. For built-in accounts on the system, use the trustee name without a domain.
trustee_sid	oval-def:EntityStateStringType (0..1)	The trustee_sid entity identifies the SID that has been granted the specified user right/privilege.

< volume_test >

The volume_test is used to check information about different storage volumes found on a Windows system. This includes the various system flags returned by GetVolumeInformation(). It is important to note that these system flags are specific to certain versions of Windows. As a result, the documentation for that version of Windows should be consulted for more information. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a volume_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 728: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< volume_object >

The volume_object element is used by a volume test to define the specific volume(s) to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A volume object defines the rootpath of the volume(s).

Extends: oval-def:ObjectType

Child Elements

Table 729: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
rootpath	oval-def:EntityObjectStringType (1..1)	A string that contains the root directory of the volume to be described. A trailing backslash is required. For example, you would specify \MyServerMyShare as “\MyServerMyShare”, or the C drive as “C:”.
oval-def:filter	n/a (0..unbounded)	

< volume_state >

The volume_state element defines the different metadata associate with a storage volume in Windows. This includes the rootpath, the file system type, name, and serial number, as well as any associated flags. Please refer to the individual elements in the schema for more details about what each represents. The GetVolumeInformation function as defined by Microsoft is also a good place to look for information.

Extends: oval-def:StateType

Child Elements

Table 730: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
root-path	oval-def:EntityStateStringType (0..1)	A string that contains the root directory of the volume to be described. A trailing backslash is required. For example, you would specify \MyServerMyShare as "\MyServerMyShare", or the C drive as "C:".
file_system	oval-def:EntityStateStringType (0..1)	The type of filesystem. For example FAT or NTFS.
name	oval-def:EntityStateStringType (0..1)	The name of the volume.
drive_type	win-def:EntityStateDriveTypeType (0..1)	The drive type of the volume.
volume_max_component_length	oval-def:EntityStateIntType (0..1)	The volume_max_component_length element specifies the maximum length, in TCHARs, of a file name component that a specified file system supports. A file name component is the portion of a file name between backslashes. The value that is stored in the variable that *lpMaximumComponentLength points to is used to indicate that a specified file system supports long names. For example, for a FAT file system that supports long names, the function stores the value 255, rather than the previous 8.3 indicator. Long names can also be supported on systems that use the NTFS file system.
serial_number	oval-def:EntityStateIntType (0..1)	The volume serial number.
file_case_sensitive_search	oval-def:EntityStateBoolType (0..1)	The file system supports case-sensitive file names.
file_case_preserved_names	oval-def:EntityStateBoolType (0..1)	The file system preserves the case of file names when it places a name on disk.
file_unicode_on_disk	oval-def:EntityStateBoolType (0..1)	The file system supports Unicode in file names as they appear on disk.
file_persistent_acls	oval-def:EntityStateBoolType (0..1)	The file system preserves and enforces ACLs. For example, NTFS preserves and enforces ACLs, and FAT does not.
file_file_compression	oval-def:EntityStateBoolType (0..1)	The file system supports file-based compression.
file_volume_quotas	oval-def:EntityStateBoolType (0..1)	The file system supports disk quotas.
file_supports_sparse_files	oval-def:EntityStateBoolType (0..1)	The file system supports sparse files.
file_supports_reparse_points	oval-def:EntityStateBoolType (0..1)	The file system supports reparse points.
file_supports_remote_storage	oval-def:EntityStateBoolType (0..1)	The file system supports remote storage.
file_volume_is_compressed	oval-def:EntityStateBoolType (0..1)	The specified volume is a compressed volume; for example, a DoubleSpace volume.

< wmi_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.7
- Reason: Replaced by the wmi57_test. This test only allows for single fields to be selected from WMI. A new test was created to allow more than one field to be selected in one statement. See the wmi57_test.
- Comment: This test has been deprecated and may be removed in a future version of the language.

The wmi test is used to check information accessed by WMI. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a wmi_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 731: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< wmi_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.7
- Reason: Replaced by the wmi57_object. This object allows for single fields to be selected from WMI. A new object was created to allow more than one field to be selected in one statement. See the wmi57_object.
- Comment: This object has been deprecated and may be removed in a future version of the language.

Extends: oval-def:ObjectType

Child Elements

Table 732: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
namespace	oval-def:EntityObjectStringType (1..1)	Specifies which WMI namespace to look under. Each WMI provider normally registers its own WMI namespace and then all its classes within that namespace. For example, all Win32 WMI classes can be found in the namespace “rootcimv2”, all IIS WMI classes can be found at “rootmicrosoftiisv2”, and all LDAP WMI classes can be found at “rootdirectoryldap”.
wql	oval-def:EntityObjectStringType (1..1)	A WQL query used to identify the object(s) to test against. Any valid WQL query is usable with one exception, at most one field is allowed in the SELECT portion of the query. For example SELECT name FROM … is valid, as is SELECT ‘true’ FROM …, but SELECT name, number FROM … is not valid. This is because the result element in the data section is only designed to work against a single field.

< wmi_state > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.7
- Reason: Replaced by the wmi57_state. This object allows for single fields to be selected from WMI. A new state was created to allow more than one field to be selected in one statement. See the wmi57_state.
- Comment: This state has been deprecated and may be removed in a future version of the language.

Extends: oval-def:StateType

Child Elements

Table 733: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
namespace	oval-def:EntityStateStringType (0..1)	Specifies which WMI namespace to look under. Each WMI provider normally registers its own WMI namespace and then all its classes within that namespace. For example, all Win32 WMI classes can be found in the namespace “rootcimv2”, all IIS WMI classes can be found at “rootmicrosoftiisv2”, and all LDAP WMI classes can be found at “rootdirectoryldap”.
wql	oval-def:EntityStateStringType (0..1)	A WQL query used to identify the object(s) to test against. Any valid WQL query is usable with one exception, at most one field is allowed in the SELECT portion of the query. For example SELECT name FROM … is valid, as is SELECT ‘true’ FROM …, but SELECT name, number FROM … is not valid. This is because the result element in the data section is only designed to work against a single field.
result	oval-def:EntityStateAnySimpleType (0..1)	The result element specifies how to test objects in the result set of the specified WQL statement. Only one comparable field is allowed. So if the WQL statement look like ‘SELECT name FROM …’, then a result element with a value of ‘Fred’ would test that value against the names returned by the WQL statement.

< wmi57_test >

The wmi57 test is used to check information accessed by WMI. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a wmi57_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 734: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< wmi57_object >

Extends: oval-def:ObjectType

Child Elements

Table 735: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
namespace	oval-def:EntityObjectStringType (1..1)	Specifies which WMI namespace to look under. Each WMI provider normally registers its own WMI namespace and then all its classes within that namespace. For example, all Win32 WMI classes can be found in the namespace “rootcimv2”, all IIS WMI classes can be found at “rootmicrosoftiisv2”, and all LDAP WMI classes can be found at “rootdirectoryldap”.
wql	oval-def:EntityObjectStringType (1..1)	A WQL query used to identify the object(s) to test against. Any valid WQL query is usable with one exception, all fields must be named in the SELECT portion of the query. For example SELECT name, age FROM … is valid. However, SELECT * FROM … is not valid. This is because the record element in the state and item require a unique field name value to ensure that any query results can be evaluated consistently.
oval-def:filter	n/a (0..unbounded)	

< wmi57_state >

Extends: oval-def:StateType

Child Elements

Table 736: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
namespace	oval-def:EntityStateStringType (0..1)	Specifies which WMI namespace to look under. Each WMI provider normally registers its own WMI namespace and then all its classes within that namespace. For example, all Win32 WMI classes can be found in the namespace “rootcimv2”, all IIS WMI classes can be found at “rootmicrosoftiisv2”, and all LDAP WMI classes can be found at “rootdirectoryldap”.
wql	oval-def:EntityStateStringType (0..1)	A WQL query used to identify the object(s) to test against. Any valid WQL query is usable with one exception, all fields must be named in the SELECT portion of the query. For example SELECT name, age FROM … is valid. However, SELECT * FROM … is not valid. This is because the record element in the state and item require a unique field name value to ensure that any query results can be evaluated consistantly.
result	oval-def:EntityStateRecordType (0..1)	The result element specifies how to test items in the result set of the specified WQL statement.

< wuaupdatesearcher_test >

The wuaupdatesearcher_test is used to evaluate patch level in a Windows environment utilizing the WUA (Windows Update Agent) interface. It is based on the Search method of the IUpdateSearcher interface found in the WUA API. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a wuaupdatesearcher_object and the optional state element specifies the metadata to check.

Note that WUA can work off of many different sources including WSUS, update.microsoft.com, and a local cab file. The content source is specific to a given system evaluating a wuaupdatesearcher_test and thus is not defined by this test. The tool being used for evaluation should determine what content source is best for the system being assessed and then evaluate this test based on that selection.

Extends: oval-def:TestType

Child Elements

Table 737: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< wuaupdatesearcher_object >

The wuaupdatesearcher_object element is used by a wuaupdatesearcher_test to define the specific search criteria to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 738: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	win-def:WuaUpdateSearcherBehaviors (0..1)	
search_criteria	oval-def:EntityObjectStringType (1..1)	The search_criteria entity specifies a search criteria to use when generating a search result. The string used for the search criteria entity must match the custom search language for Search method of the IUpdateSearcher interface. The string consists of criteria that are evaluated to determine which updates to return. The Search method performs a synchronous search for updates by using the current configured search options. For more information about possible search criteria, please see the Search method of the IUpdateSearcher interface.
oval-def:filter	n/a (0..unbounded)	

< wuaupdatesearcher_state >

The wuaupdatesearcher_state element defines entities that can be tested related to a uaupdatesearcher_object. This includes the search criteria and updated id. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 739: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
search_criteria	oval-def:EntityStateStringType (0..1)	The search_criteria entity specifies a string to examine the search criteria that was used to generate the object set. Note that since this entity is part of the state, it is not used to determine the object set, but rather is used to test the search criteria that was actually used.
update_id	oval-def:EntityStateStringType (0..1)	The update_id enity specifies a string that represents a revision-independent identifier of an update. This information is part of the IUpdateIdentity interface that is part of the result of the IUpdateSearcher interface's Search method.

== WuaUpdateSearcherBehaviors ==

The WuaUpdateSearcherBehaviors complex type defines behaviors that allow a more detailed definition of the wuaupdatesearcher_object being specified. 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.

Attributes

Table 740: Attributes

Attribute	Type	Desc.
include_superseded_updates	xsd:boolean (optional *default*=’true’)	‘include_superseded_updates’ is a boolean flag that when set to true indicates that the search results should include updates that are superseded by other updates in the search results. When set to ‘false’ superseded updates should be excluded from the set of matching update items. The default value is ‘true’.

== EntityStateAddrTypeType ==

The EntityStateAddrTypeType complex type restricts a string value to a specific set of values that describe address types associated with an interface. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 741: Enumeration Values

Value	Description
MIB_IPADDR_DELETED	The stated IP address is being deleted. The unsigned short value that this corresponds to is 0x0040
MIB_IPADDR_DISCONNECTED	The stated IP address is on a disconnected interface. The unsigned short value that this corresponds to is 0x0008.
MIB_IPADDR_DYNAMIC	The stated IP address is a dynamic IP address. The unsigned short value that this corresponds to is 0x0004.
MIB_IPADDR_PRIMARY	The stated IP address is a primary IP address. The unsigned short value that this corresponds to is 0x0001.
MIB_IPADDR_TRANSIENT	The stated IP address is a transient IP address. The unsigned short value that this corresponds to is 0x0080
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateAdstypeType ==

The EntityStateAdstypeType complex type restricts a string value to a specific set of values that specify the different types of information that an active directory attribute can represents. For more information look at the ADSTYPE-ENUM enumeration defined by Microsoft. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 742: Enumeration Values

Value	Description
ADSTYPE_INVALID	The data type is invalid.
ADSTYPE_DN_STRING	The string is of Distinguished Name (path) of a directory service object.

Continued on next page

Table 742 – continued from previous page

Value	Description
ADSTYPE_CASE_EXACT_STRING	The string is of the case-sensitive type.
ADSTYPE_CASE_IGNORE_STRING	The string is of the case-insensitive type.
ADSTYPE_PRINTABLE_STRING	The string is displayable on the screen or in print.
ADSTYPE_NUMERIC_STRING	The string is of a numeric value to be interpreted as text.
ADSTYPE_BOOLEAN	The data is of a Boolean value.
ADSTYPE_INTEGER	The data is of an integer value.
ADSTYPE_OCTET_STRING	The string is of a byte array.
ADSTYPE_UTC_TIME	The data is of the universal time as expressed in Universal Time Coordinate (UTC).
ADSTYPE_LARGE_INTEGER	The data is of a long integer value.
ADSTYPE_PROV_SPECIFIC	The string is of a provider-specific string.
ADSTYPE_OBJECT_CLASS	Not used.
ADSTYPE_CASEIGNORE_LIST	The data is of a list of case insensitive strings.
ADSTYPE_OCTET_LIST	The data is of a list of octet strings.
ADSTYPE_PATH	The string is of a directory path.
ADSTYPE_POSTALADDRESS	The string is of the postal address type.
ADSTYPE_TIMESTAMP	The data is of a time stamp in seconds.

Continued on next page

Table 742 – continued from previous page

Value	Description
ADSTYPE_BACKLINK	The string is of a back link.
ADSTYPE_TYPEDNAME	The string is of a typed name.
ADSTYPE_HOLD	The data is of the Hold data structure.
ADSTYPE_NETADDRESS	The string is of a net address.
ADSTYPE_REPLICAPOINTER	The data is of a replica pointer.
ADSTYPE_FAXNUMBER	The string is of a fax number.
ADSTYPE_EMAIL	The data is of an e-mail message.
ADSTYPE_NT_SECURITY_DESCRIPTOR	The data is of Windows NT/Windows 2000 Security Descriptor as represented by a byte array.
ADSTYPE_UNKNOWN	The data is of an undefined type.
ADSTYPE_DN_WITH_BINARY	The data is of ADS_DN_WITH_BINARY used for mapping a distinguished name to a non varying GUID.
ADSTYPE_DN_WITH_STRING	The data is of ADS_DN_WITH_STRING used for mapping a distinguished name to a non-varying string value.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateAuditType ==

The EntityStateAuditType complex type restricts a string value to a specific set of values: AUDIT_NONE, AUDIT_SUCCESS, AUDIT_FAILURE, and AUDIT_SUCCESS_FAILURE. These values describe which audit records should be generated. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 743: Enumeration Values

Value	Description
AUDIT_FAILURE	The audit type AUDIT_FAILURE is used to perform audits on all unsuccessful occurrences of specified events when auditing is enabled.
AUDIT_NONE	The audit type AUDIT_NONE is used to cancel all auditing options for the specified events.
AUDIT_SUCCESS	The audit type AUDIT_SUCCESS is used to perform audits on all successful occurrences of the specified events when auditing is enabled.
AUDIT_SUCCESS_FAILURE	The audit type AUDIT_SUCCESS_FAILURE is used to perform audits on all successful and unsuccessful occurrences of the specified events when auditing is enabled.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateDriveTypeType ==

The EntityStateDriveTypeType complex type defines the different values that are valid for the drive_type entity of a win-def:volume_state. Note that the Windows API returns a UINT value and OVAL uses the constant name that is normally defined for these return values. This is done to increase readability and maintainability of OVAL Definitions. The empty string is also allowed as a valid value to support an empty element that is found when a variable reference is used within the drive_type entity. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 744: Enumeration Values

Value	Description
DRIVE_UNKNOWN	The DRIVE_UNKNOWN type means that drive type cannot be determined. The UINT value that this corresponds to is 0.
DRIVE_NO_ROOT_DIR	The DRIVE_NO_ROOT_DIR type means that the root path is not valid. The UINT value that this corresponds to is 1.
DRIVE_REMOVABLE	The DRIVE_REMOVABLE type means that the drive contains removable media. The UINT value that this corresponds to is 2.
DRIVE_FIXED	The DRIVE_FIXED type means that the drive contains fixed media. The UINT value that this corresponds to is 3.
DRIVE_REMOTE	The DRIVE_REMOTE type means that the drive is a remote drive (i.e. network drive). The UINT value that this corresponds to is 4.
DRIVE_CDROM	The DRIVE_CDROM type means that the drive is a CD-ROM drive. The UINT value that this corresponds to is 5.
DRIVE_RAMDISK	The DRIVE_RAMDISK type means that the drive is a RAM disk. The UINT value that this corresponds to is 6.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateInterfaceTypeType ==

The EntityStateInterfaceTypeType complex type restricts a string value to a specific set of values. These values describe the different interface types. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 745: Enumeration Values

Value	Description
MIB_IF_TYPE_ETHERNET	The MIB_IF_TYPE_ETHERNET type is used to describe ethernet interfaces.
MIB_IF_TYPE_FDDI	The MIB_IF_TYPE_FDDI type is used to describe fiber distributed data interfaces (FDDI).
MIB_IF_TYPE_LOOPBACK	The MIB_IF_TYPE_LOOPBACK type is used to describe loopback interfaces.
MIB_IF_TYPE_OTHER	The MIB_IF_TYPE_OTHER type is used to describe unknown interfaces.
MIB_IF_TYPE_PPP	The MIB_IF_TYPE_PPP type is used to describe point-to-point protocol interfaces (PPP).
MIB_IF_TYPE_SLIP	The MIB_IF_TYPE_SLIP type is used to describe serial line internet protocol interfaces (SLIP).
MIB_IF_TYPE_TOKENRING	The MIB_IF_TYPE_TOKENRING type is used to describe token ring interfaces..
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateFileTypeType ==

The EntityStateFileTypeType complex type restricts a string value to a specific set of values. These values describe the type of file being represented. For more information see the GetFileType and GetFileAttributesEx functions as defined by Microsoft. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 746: Enumeration Values

Value	Description
FILE_ATTRIBUTE_DIRECTORY (Deprecated)	The handle identifies a directory. **Deprecated As Of Version:** 5.11.1:1.2 **Reason:** In version 5.11.1:1.2 of the OVAL Language windows schema, a file_attributes entity was added to the file_state, obviating the need to overload this attribute with the file-type enumeration. **Comment:** This value has been deprecated and will be removed in version 6.0 of the language.
FILE_TYPE_CHAR	The specified file is a character file, typically an LPT device or a console.
FILE_TYPE_DISK	The specified file is a disk file.
FILE_TYPE_PIPE	The specified file is a socket, a named pipe, or an anonymous pipe.
FILE_TYPE_REMOTE	Unused.
FILE_TYPE_UNKNOWN	Either the type of the specified file is unknown, or the function failed.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateFileAttributeType ==

The EntityStateFileAttributeType complex type restricts a string value to a specific set of values. These values describe the Windows file attribute being represented. For more information see the GetFileAttributes and GetFileAttributesEx functions as defined by Microsoft. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 747: Enumeration Values

Value	Description
FILE_ATTRIBUTE_ARCHIVE	A file or directory that is an archive file or directory. Applications typically use this attribute to mark files for backup or removal.
FILE_ATTRIBUTE_COMPRESSED	A file or directory that is compressed. For a file, all of the data in the file is compressed. For a directory, compression is the default for newly created files and subdirectories.
FILE_ATTRIBUTE_DEVICE	This value is reserved for system use.
FILE_ATTRIBUTE_DIRECTORY	The handle that identifies a directory.
FILE_ATTRIBUTE_ENCRYPTED	A file or directory that is encrypted. For a file, all data streams in the file are encrypted. For a directory, encryption is the default for newly created files and subdirectories.
FILE_ATTRIBUTE_HIDDEN	The file or directory is hidden. It is not included in an ordinary directory listing.
FILE_ATTRIBUTE_INTEGRITY_STREAM	The directory or user data stream is configured with integrity (only supported on ReFS volumes). It is not included in an ordinary directory listing. The integrity setting persists with the file if it's renamed. If a file is copied the destination file will have integrity set if either the source file or destination directory have integrity set.Windows Server 2008 R2, Windows 7, Windows Server 2008, Windows Vista, Windows Server 2003, and Windows XP: This flag is not supported until Windows Server 2012.
FILE_ATTRIBUTE_NORMAL	A file that does not have other attributes set. This attribute is valid only when used alone.
FILE_ATTRIBUTE_NOT_CONTENT_INDEXED	The file or directory is not to be indexed by the content indexing service.
FILE_ATTRIBUTE_NO_SCRUB_DATA	The user data stream not to be read by the background data integrity scanner (AKA scrubber). When set on a directory it only provides inheritance. This flag is only

== EntityObjectNamingContextType ==

The EntityObjectNamingContextType restricts a string value to a specific set of values: domain, configuration, and schema. These values describe the different default naming context found in active directory. A naming context is defined as a single object in the Directory Information Tree (DIT) along with every object in the tree subordinate to it. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityObjectStringType

Table 748: Enumeration Values

Value	Description
domain	The domain naming context contains Active Directory objects present in the specified domain (e.g. users, computers, groups, and other objects).
configuration	The configuration naming context contains configuration data that is required for the Active Directory to operate as a directory service.
schema	The schema naming context contains all of the Active Directory object definitions.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateNamingContextType ==

The EntityStateNamingContextType restricts a string value to a specific set of values: domain, configuration, and schema. These values describe the different default naming context found in active directory. A naming context is defined as a single object in the Directory Information Tree (DIT) along with every object in the tree subordinate to it. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 749: Enumeration Values

Value	Description
domain	The domain naming context contains Active Directory objects present in the specified domain (e.g. users, computers, groups, and other objects).
configuration	The configuration naming context contains configuration data that is required for the Active Directory to operate as a directory service.
schema	The schema naming context contains all of the Active Directory object definitions.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateNTUserAccountTypeType ==

The EntityStateNTUserAccountTypeType restricts a string value to a specific set of values that describe the different types of accounts. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 750: Enumeration Values

Value	Description
local	Local accounts are accounts that were created directly on the machine being tested and should be in the form of machinenameusername
domain	Domain accounts are accounts that were created on a domain controller and should be in the form of domainusername
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStatePeTargetMachineType ==

The EntityStatePeTargetMachineType enumeration identifies the valid machine targets that can be specified in the PE file header. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 751: Enumeration Values

Value	Description
IMAGE_FILE_MACHINE_UNKNOWN	The IMAGE_FILE_MACHINE_UNKNOWN type is used to indicate an unknown machine.
IMAGE_FILE_MACHINE_ALPHA	The IMAGE_FILE_MACHINE_ALPHA type is used to indicate an Alpha APX machine.
IMAGE_FILE_MACHINE_ARM	The IMAGE_FILE_MACHINE_ARM type is used to indicate an ARM little endian machine.
IMAGE_FILE_MACHINE_ALPHA64	The IMAGE_FILE_MACHINE_ALPHA64 type is used to indicate an 64-bit Alpha APX machine.
IMAGE_FILE_MACHINE_I386	The IMAGE_FILE_MACHINE_I386 type is used to indicate an Intel 386 machine.
IMAGE_FILE_MACHINE_IA64	The IMAGE_FILE_MACHINE_IA64 type is used to indicate an Intel Itanium machine.
IMAGE_FILE_MACHINE_M68K	The IMAGE_FILE_MACHINE_M68K type is used to indicate an M68K machine.
IMAGE_FILE_MACHINE_MIPS16	The IMAGE_FILE_MACHINE_MIPS16 type is used to indicate a MIPS16 machine.
IMAGE_FILE_MACHINE_MIPSFPU	The IMAGE_FILE_MACHINE_MIPSFPU type is used to indicate an MIPS machine with FPU.
IMAGE_FILE_MACHINE_MIPSFPU16	The IMAGE_FILE_MACHINE_MIPSFPU16 type is used to indicate a MIPS16 machine with FPU.
IMAGE_FILE_MACHINE_POWERPC	The IMAGE_FILE_MACHINE_POWERPC type is used to indicate an Power PC little endian machine.
IMAGE_FILE_MACHINE_R3000	The IMAGE_FILE_MACHINE_R3000 type is used to indicate a MIPS little endian, 0x160 big endian machine.
IMAGE_FILE_MACHINE_R4000	

== EntityStatePeSubsystemType ==

The EntityStatePeSubsystemType enumeration identifies the valid subsystem types that can be specified in the PE file header. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 752: Enumeration Values

Value	Description
IMAGE_SUBSYSTEM_UNKNOWN	The IMAGE_SUBSYSTEM_UNKNOWN type is used to indicate an unknown subsystem.
IMAGE_SUBSYSTEM_NATIVE	The IMAGE_SUBSYSTEM_NATIVE type is used to indicate that no subsystem is required.
IMAGE_SUBSYSTEM_WINDOWS_GUI	The IMAGE_SUBSYSTEM_WINDOWS_GUI type is used to indicate a Windows graphical user interface (GUI) subsystem.
IMAGE_SUBSYSTEM_WINDOWS_CUI	The IMAGE_SUBSYSTEM_WINDOWS_CUI type is used to indicate a Windows character-mode user interface (CUI) subsystem.
IMAGE_SUBSYSTEM_OS2_CUI	The IMAGE_SUBSYSTEM_OS2_CUI type is used to indicate an OS/2 CUI subsystem.
IMAGE_SUBSYSTEM_POSIX_CUI	The IMAGE_SUBSYSTEM_POSIX_CUI type is used to indicate a POSIX CUI subsystem.
IMAGE_SUBSYSTEM_WINDOWS_CE_GUI	The IMAGE_SUBSYSTEM_WINDOWS_CE_GUI type is used to indicate a Windows CE system.
IMAGE_SUBSYSTEM_EFI_APPLICATION	The IMAGE_SUBSYSTEM_EFI_APPLICATION type is used to indicate an Extensible Firmware Interface (EFI) application.
IMAGE_SUBSYSTEM_EFI_BOOT_SERVICE_DRIVER	The IMAGE_SUBSYSTEM_EFI_BOOT_SERVICE_DRIVER type is used to indicate a EFI driver with boot services.
IMAGE_SUBSYSTEM_EFI_RUNTIME_DRIVER	The IMAGE_SUBSYSTEM_EFI_RUNTIME_DRIVER type is used to indicate a EFI driver with run-time services subsystem.
IMAGE_SUBSYSTEM_EFI_ROM	The IMAGE_SUBSYSTEM_EFI_ROM type is used to indicate an EFI ROM image.
IMAGE_SUBSYSTEM_XBOX	

== EntityObjectProtocolType ==

The EntityObjectProtocolType restricts a string value to a specific set of values: TCP and UDP. These values describe the different protocols available to a port. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityObjectStringType

Table 753: Enumeration Values

Value	Description
TCP	The port uses the Transmission Control Protocol (TCP).
UDP	The port uses the User Datagram Protocol (UDP).
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateProtocolType ==

The EntityStateProtocolType restricts a string value to a specific set of values: TCP and UDP. These values describe the different protocols available to a port. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 754: Enumeration Values

Value	Description
TCP	The port uses the Transmission Control Protocol (TCP).
UDP	The port uses the User Datagram Protocol (UDP).
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityObjectRegistryHiveType ==

The EntityObjectRegistryHiveType restricts a string value to a specific set of values: HKEY_CLASSES_ROOT, HKEY_CURRENT_CONFIG, HKEY_CURRENT_USER, HKEY_CURRENT_USER_LOCAL_SETTINGS, HKEY_LOCAL_MACHINE, and HKEY_USERS. These values describe the possible hives in the registry. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityObjectStringType

Table 755: Enumeration Values

Value	Description
HKEY_CLASSES_ROOT	This registry subtree contains information that associates file types with programs and configuration data for automation (e.g. COM objects and Visual Basic Programs).
HKEY_CURRENT_CONFIG	This registry subtree contains configuration data for the current hardware profile.
HKEY_CURRENT_USER	This registry subtree contains the user profile of the user that is currently logged into the system.
HKEY_CURRENT_USER_LOCAL_SETTINGS	Registry entries subordinate to this key define preferences of the current user that are local to the machine. These entries are not included in the per-user registry portion of a roaming user profile. This key is supported starting with Windows 7 and Windows Server 2008 R2.
HKEY_LOCAL_MACHINE	This registry subtree contains information about the local system.
HKEY_USERS	This registry subtree contains user-specific data.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateRegistryHiveType ==

The EntityStateRegistryHiveType restricts a string value to a specific set of values: HKEY_CLASSES_ROOT, HKEY_CURRENT_CONFIG, HKEY_CURRENT_USER, HKEY_LOCAL_MACHINE, and HKEY_USERS. These values describe the possible hives in the registry. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 756: Enumeration Values

Value	Description
HKEY_CLASSES_ROOT	This registry subtree contains information that associates file types with programs and configuration data for automation (e.g. COM objects and Visual Basic Programs).
HKEY_CURRENT_CONFIG	This registry subtree contains configuration data for the current hardware profile.
HKEY_CURRENT_USER	This registry subtree contains the user profile of the user that is currently logged into the system.
HKEY_LOCAL_MACHINE	This registry subtree contains information about the local system.
HKEY_USERS	This registry subtree contains user-specific data.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateRegistryTypeType ==

The EntityStateRegistryTypeType complex type defines the different values that are valid for the type entity of a registry state. These values describe the possible types of data stored in a registry key. The empty string is also allowed as a valid value to support an empty element that is found when a variable reference is used within the type entity. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values. Please note that the values identified are for the type entity and are not valid values for the datatype attribute. For information about how to encode registry data in OVAL for each of the different types, please visit the registry_state documentation.

Restricts: oval-def:EntityStateStringType

Table 757: Enumeration Values

Value	Description
reg_binary	The reg_binary type is used by registry keys that specify binary data in any form.
reg_dword	The reg_dword type is used by registry keys that specify an unsigned 32-bit integer.
reg_dword_little_endian (Deprecated)	The reg_dword_little_endian type is used by registry keys that specify an unsigned 32-bit little-endian integer. It is designed to run on little-endian computer architectures. **Deprecated As Of Version:** 5.11.1:1.1 **Reason:** Defined to have same value as reg_dword. **Comment:** This registry type enumeration value has been deprecated and may be removed in a future version of the language.
reg_dword_big_endian	The reg_dword_big_endian type is used by registry keys that specify an unsigned 32-bit big-endian integer. It is designed to run on big-endian computer architectures.
reg_expand_sz	The reg_expand_sz type is used by registry keys to specify a null-terminated string that contains unexpanded references to environment variables (for example, “%PATH%”).
reg_link	The reg_link type is used by the registry keys for null-terminated unicode strings. It is related to target path of a symbolic link created by the RegCreateKeyEx function.
reg_multi_sz	The reg_multi_sz type is used by registry keys that specify an array of null-terminated strings, terminated by two null characters.
reg_none	The reg_none type is used by registry keys that have no defined value type.
reg_qword	The reg_qword type is used by registry keys that specify an unsigned 64-bit integer.
reg_qword_little_endian (Deprecated)	

== EntityStateServiceControlsAcceptedType ==

The EntityStateServiceAcceptedControlsType complex type defines the different values that are valid for the controls_accepted entity of a service. Note that the Windows API returns a DWORD value and OVAL uses the constant name that is normally defined for these return values. This is done to increase readability and maintainability of OVAL Definitions. The empty string is also allowed as a valid value to support an empty element that is found when a variable reference is used within the controls_accepted entity. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 758: Enumeration Values

Value	Description
SERVICE_ACCEPT_NETBINDCHANGE	The SERVICE_ACCEPT_NETBINDCHANGE type means that the service is a network component and can accept changes in its binding without being stopped or restarted. The DWORD value that this corresponds to is 0x00000010.
SERVICE_ACCEPT_PARAMCHANGE	The SERVICE_ACCEPT_PARAMCHANGE type means that the service can re-read its startup parameters without being stopped or restarted. The DWORD value that this corresponds to is 0x00000008.
SERVICE_ACCEPT_PAUSE_CONTINUE	The SERVICE_ACCEPT_PAUSE_CONTINUE type means that the service can be paused or continued. The DWORD value that this corresponds to is 0x00000002.
SERVICE_ACCEPT_PRESHUTDOWN	The SERVICE_ACCEPT_PRESHUTDOWN type means that the service can receive pre-shutdown notifications. The DWORD value that this corresponds to is 0x00000100.
SERVICE_ACCEPT_SHUTDOWN	The SERVICE_ACCEPT_SHUTDOWN type means that the service can receive shutdown notifications. The DWORD value that this corresponds to is 0x00000004.
SERVICE_ACCEPT_STOP	The SERVICE_ACCEPT_STOP type means that the service can be stopped. The DWORD value that this corresponds to is 0x00000001.
SERVICE_ACCEPT_HARDWAREPROFILECHANGE	The SERVICE_ACCEPT_HARDWAREPROFILECHANGE type means that the service can receive notifications when the system's hardware profile changes. The DWORD value that this corresponds to is 0x00000020.
SERVICE_ACCEPT_POWEREVENT	The SERVICE_ACCEPT_POWEREVENT type means that the service can receive notifications when the system's power status has changed. The DWORD value that this corresponds to is 0x00000040.
SERVICE_ACCEPT_SESSIONCHANGE	The SERVICE_ACCEPT_SESSIONCHANGE type means that the service can receive notifications when the system's session status has changed. The DWORD value that this corresponds to is 0x00000080.

== EntityStateServiceCurrentStateType ==

The EntityStateServiceCurrentStateType complex type defines the different values that are valid for the current_state entity of a service. Note that the Windows API returns a DWORD value and OVAL uses the constant name that is normally defined for these return values. This is done to increase readability and maintainability of OVAL Definitions. The empty string is also allowed as a valid value to support an empty element that is found when a variable reference is used within the current_state entity. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 759: Enumeration Values

Value	Description
SERVICE_CONTINUE_PENDING	The SERVICE_CONTINUE_PENDING type means that the service has been sent a command to continue, however, the command has not yet been executed. The DWORD value that this corresponds to is 0x00000005.
SERVICE_PAUSE_PENDING	The SERVICE_PAUSE_PENDING type means that the service has been sent a command to pause, however, the command has not yet been executed. The DWORD value that this corresponds to is 0x00000006.
SERVICE_PAUSED	The SERVICE_PAUSED type means that the service is paused. The DWORD value that this corresponds to is 0x00000007.
SERVICE_RUNNING	The SERVICE_RUNNING type means that the service is running. The DWORD value that this corresponds to is 0x00000004.
SERVICE_START_PENDING	The SERVICE_START_PENDING type means that the service has been sent a command to start, however, the command has not yet been executed. The DWORD value that this corresponds to is 0x00000002.
SERVICE_STOP_PENDING	The SERVICE_STOP_PENDING type means that the service has been sent a command to stop, however, the command has not yet been executed. The DWORD value that this corresponds to is 0x00000003.
SERVICE_STOPPED	The SERVICE_STOPPED type means that the service is stopped. The DWORD value that this corresponds to is 0x00000001.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateServiceStartTypeType ==

The EntityStateServiceStartTypeType complex type defines the different values that are valid for the start_type entity of a service. Note that the Windows API returns a DWORD value and OVAL uses the constant name that is normally defined for these return values. This is done to increase readability and maintainability of OVAL Definitions. The empty string is also allowed as a valid value to support an empty element that is found when a variable reference is used within the start_type entity. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 760: Enumeration Values

Value	Description
SERVICE_AUTO_START	The SERVICE_AUTO_START type means that the service is started automatically by the Service Control Manager (SCM) during startup. The DWORD value that this corresponds to is 0x00000002.
SERVICE_BOOT_START	The SERVICE_BOOT_START type means that the driver service is started by the system loader. The DWORD value that this corresponds to is 0x00000000.
SERVICE_DEMAND_START	The SERVICE_DEMAND_START type means that the service is started by the Service Control Manager (SCM) when StartService() is called. The DWORD value that this corresponds to is 0x00000003.
SERVICE_DISABLED	The SERVICE_DISABLED type means that the service cannot be started. The DWORD value that this corresponds to is 0x00000004.
SERVICE_SYSTEM_START	The SERVICE_SYSTEM_START type means that the service is a device driver started by IoInitSystem(). The DWORD value that this corresponds to is 0x00000001.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateServiceTypeType ==

The EntityStateServiceTypeType complex type defines the different values that are valid for the service_type entity of a service. Note that the Windows API returns a DWORD value and OVAL uses the constant name that is normally defined for these return values. This is done to increase readability and maintainability of OVAL Definitions. The empty string is also allowed as a valid value to support an empty element that is found when a variable reference is used within the service_type entity. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 761: Enumeration Values

Value	Description
SERVICE_FILE_SYSTEM_DRIVER	The SERVICE_FILE_SYSTEM_DRIVER type means that the service is a file system driver. The DWORD value that this corresponds to is 0x00000002.
SERVICE_KERNEL_DRIVER	The SERVICE_KERNEL_DRIVER type means that the service is a driver. The DWORD value that this corresponds to is 0x00000001.
SERVICE_WIN32_OWN_PROCESS	The SERVICE_WIN32_OWN_PROCESS type means that the service runs in its own process. The DWORD value that this corresponds to is 0x00000010.
SERVICE_WIN32_SHARE_PROCESS	The SERVICE_WIN32_SHARE_PROCESS type means that the service runs in a process with other services. The DWORD value that this corresponds to is 0x00000020.
SERVICE_INTERACTIVE_PROCESS	The SERVICE_WIN32_SHARE_PROCESS type means that the service runs in a process with other services. The DWORD value that this corresponds to is 0x00000100.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateSharedResourceTypeType ==

The EntityStateSharedResourceTypeType complex type defines the different values that are valid for the type entity of a shared resource state. Note that the Windows API returns a DWORD value and OVAL uses the constant name that is normally defined for these return values. This is done to increase readability and maintainability of OVAL Definitions. The empty string is also allowed as a valid value to support an empty element that is found when a variable reference is used within the type entity. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

It is also important to note that special shared resources are those reserved for remote administration, interprocess communication, and administrative shares.

Restricts: oval-def:EntityStateStringType

Table 762: Enumeration Values

Value	Description
STYPE_DISKTREE	The STYPE_DISKTREE type means that the shared resource is a disk drive. The DWORD value that this corresponds to is 0x00000000.
STYPE_DISKTREE_SPECIAL	The STYPE_DISKTREE_SPECIAL type means that the shared resource is a special disk drive. The DWORD value that this corresponds to is 0x80000000.
STYPE_DISKTREE_TEMPORARY	The STYPE_DISKTREE_TEMPORARY type means that the shared resource is a temporary disk drive. The DWORD value that this corresponds to is 0x40000000.
STYPE_DISKTREE_SPECIAL_TEMPORARY	The STYPE_DISKTREE_SPECIAL_TEMPORARY type means that the shared resource is a temporary, special disk drive. The DWORD value that this corresponds to is 0xC0000000.
STYPE_PRINTQ	The STYPE_PRINTQ type means that the shared resource is a print queue. The DWORD value that this corresponds to is 0x00000001.
STYPE_PRINTQ_SPECIAL	The STYPE_PRINTQ_SPECIAL type means that the shared resource is a special print queue. The DWORD value that this corresponds to is 0x80000001.
STYPE_PRINTQ_TEMPORARY	The STYPE_PRINTQ_TEMPORARY type means that the shared resource is a temporary print queue. The DWORD value that this corresponds to is 0x40000001.
STYPE_PRINTQ_SPECIAL_TEMPORARY	The STYPE_PRINTQ_SPECIAL_TEMPORARY type means that the shared resource is a temporary, special print queue. The DWORD value that this corresponds to is 0xC0000001.
STYPE_DEVICE	The STYPE_DEVICE type means that the shared resource is a communication device. The DWORD value that this corresponds to is 0x00000002.
STYPE_DEVICE_SPECIAL	The STYPE_DEVICE_SPECIAL type means that the shared resource is a special communication device. The DWORD value that this corresponds to is 0x80000002.

== EntityObjectSystemMetricIndexType ==

The EntityObjectSystemMetricIndexType complex type defines the different values that are valid for the index entity of a system metric object. These values describe the system metric or configuration setting to be retrieved. The empty string is also allowed as a valid value to support an empty element that is found when a variable reference is used within the index entity. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values. Please note that the values identified are for the index entity and are not valid values for the datatype attribute.

Restricts: oval-def:EntityObjectStringType

Table 763: Enumeration Values

Value	Description
SM_ARRANGE	The flags that specify how the system arranged minimized windows.
SM_CLEANBOOT	The value that specifies how the system is started.
SM_CMONITORS	The number of display monitors on a desktop.
SM_CMOUSEBUTTONS	The number of buttons on a mouse, or zero if no mouse is installed.
SM_CXBORDER	The width of a window border, in pixels. This is equivalent to the SM_CXEDGE value for windows with the 3-D look.
SM_CXCURSOR	The width of a cursor, in pixels. The system cannot create cursors of other sizes.
SM_CXDLGFRAME	This value is the same as SM_CXFIXEDFRAME.
SM_CXDOUBLECLK	The width of the rectangle around the location of a first click in a double-click sequence, in pixels.
SM_CXDRAG	The number of pixels on either side of a mouse-down point that the mouse pointer can move before a drag operation begins.
SM_CXEDGE	The width of a 3-D border, in pixels. This metric is the 3-D counterpart of SM_CXBORDER.

Continued on next page

Table 763 – continued from previous page

Value	Description
SM_CXFIXEDFRAME	The thickness of the frame around the perimeter of a window that has a caption but is not sizable, in pixels.
SM_CXFOCUSBORDER	The width of the left and right edges of the focus rectangle that the DrawFocusRect draws.
SM_CXFRAME	This value is the same as SM_CXSIZEFRAME.
SM_CXFULLSCREEN	The width of the client area for a full-screen window on the primary display monitor, in pixels.
SM_CXHSCROLL	The width of the arrow bitmap on a horizontal scroll bar, in pixels.
SM_CXHTHUMB	The width of the thumb box in a horizontal scroll bar, in pixels.
SM_CXICON	The default width of an icon, in pixels.
SM_CXICONSPACING	The width of a grid cell for items in large icon view, in pixels.
SM_CXMAXIMIZED	The default width, in pixels, of a maximized top-level window on the primary display monitor.
SM_CXMAXTRACK	The default maximum width of a window that has a caption and sizing borders, in pixels.
SM_CXMENUCHECK	The width of the default menu check-mark bitmap, in pixels.
SM_CXMENUSIZE	The width of menu bar buttons, such as the child window close button that is used in the multiple document interface, in pixels.
SM_CXMIN	The minimum width of a window, in pixels.

Continued on next page

Table 763 – continued from previous page

Value	Description
SM_CXMINIMIZED	The width of a minimized window, in pixels.
SM_CXMINSPACING	The width of a grid cell for a minimized window, in pixels.
SM_CXMINTRACK	The minimum tracking width of a window, in pixels.
SM_CXPADDEDBORDER	The amount of border padding for captioned windows, in pixels.
SM_CXSCREEN	The width of the screen of the primary display monitor, in pixels.
SM_CXSIZE	The width of a button in a window caption or title bar, in pixels.
SM_CXSIZEFRAME	The thickness of the sizing border around the perimeter of a window that can be resized, in pixels.
SM_CXSMICON	The recommended width of a small icon, in pixels.
SM_CXSMSIZE	The width of small caption buttons, in pixels.
SM_CXVIRTUALSCREEN	The width of the virtual screen, in pixels.
SM_CXVSCROLL	The width of a vertical scroll bar, in pixels.
SM_CYBORDER	The height of a window border, in pixels.
SM_CYCAPTION	The height of a caption area, in pixels.
SM_CYCURSOR	The height of a cursor, in pixels.
SM_CYDLGFRAME	This value is the same as SM_CYFIXEDFRAME.

Continued on next page

Table 763 – continued from previous page

Value	Description
SM_CYDOUBLECLK	The height of the rectangle around the location of a first click in a double-click sequence, in pixels.
SM_CYDRAG	The number of pixels above and below a mouse-down point that the mouse pointer can move before a drag operation begins.
SM_CYEDGE	The height of a 3-D border, in pixels. This is the 3-D counterpart of SM_CYBORDER.
SM_CYFIXEDFRAME	The thickness of the frame around the perimeter of a window that has a caption but is not sizable, in pixels.
SM_CYFOCUSBORDER	The height of the top and bottom edges of the focus rectangle drawn by DrawFocusRect. This value is in pixels.
SM_CYFRAME	This value is the same as SM_CYSIZEFRAME.
SM_CYFULLSCREEN	The height of the client area for a full-screen window on the primary display monitor, in pixels.
SM_CYHSCROLL	The height of a horizontal scroll bar, in pixels.
SM_CYICON	The default height of an icon, in pixels.
SM_CYICONSPACING	The height of a grid cell for items in large icon view, in pixels.
SM_CYKANJIWINDOW	For double byte character set versions of the system, this is the height of the Kanji window at the bottom of the screen, in pixels.
SM_CYMAXIMIZED	The default height, in pixels, of a maximized top-level window on the primary display monitor.

Continued on next page

Table 763 – continued from previous page

Value	Description
SM_CYMAXTRACK	The default maximum height of a window that has a caption and sizing borders, in pixels.
SM_CYMENU	The height of a single-line menu bar, in pixels.
SM_CYMENUCHECK	The height of the default menu check-mark bitmap, in pixels.
SM_CYMENUSIZE	The height of menu bar buttons, such as the child window close button that is used in the multiple document interface, in pixels.
SM_CYMIN	The minimum height of a window, in pixels.
SM_CYMINIMIZED	The height of a minimized window, in pixels.
SM_CYMINSPACING	The height of a grid cell for a minimized window, in pixels.
SM_CYMINTRACK	The minimum tracking height of a window, in pixels.
SM_CYSCREEN	The height of the screen of the primary display monitor, in pixels.
SM_CYSIZE	The height of a button in a window caption or title bar, in pixels.
SM_CYSIZEFRAME	The thickness of the sizing border around the perimeter of a window that can be resized, in pixels.
SM_CYSMCAPTION	The height of a small caption, in pixels.
SM_CYSMICON	The recommended height of a small icon, in pixels.
SM_CYSMSIZE	The height of small caption buttons, in pixels.

Continued on next page

Table 763 – continued from previous page

Value	Description
SM_CYVIRTUALSCREEN	The height of the virtual screen, in pixels. The virtual screen is the bounding rectangle of all display monitors.
SM_CYVSCROLL	The height of the arrow bitmap on a vertical scroll bar, in pixels.
SM_CYVTHUMB	The height of the thumb box in a vertical scroll bar, in pixels.
SM_DBCSENABLED	Nonzero if User32.dll supports DBCS; otherwise, 0.
SM_DEBUG	Nonzero if the debug version of User.exe is installed; otherwise, 0.
SM_DIGITIZER	Nonzero if the current operating system is Windows 7 or Windows Server 2008 R2 and the Tablet PC Input service is started; otherwise, 0. The return value is a bitmask that specifies the type of digitizer input supported by the device.
SM_IMMENABLED	Nonzero if Input Method Manager/Input Method Editor features are enabled; otherwise, 0.
SM_MAXIMUMTOUCHES	Nonzero if there are digitizers in the system; otherwise, 0.
SM_MEDIACENTER	Nonzero if the current operating system is the Windows XP, Media Center Edition, 0 if not.
SM_MENUDROPALIGNMENT	Nonzero if drop-down menus are right-aligned with the corresponding menu-bar item; 0 if the menus are left-aligned.
SM_MIDEASTENABLED	Nonzero if the system is enabled for Hebrew and Arabic languages, 0 if not.
SM_MOUSEPRESENT	Nonzero if a mouse is installed; otherwise, 0.

Continued on next page

Table 763 – continued from previous page

Value	Description
SM_MOUSEHORIZONTALWHEELPRESENT	Nonzero if a mouse with a horizontal scroll wheel is installed; otherwise 0.
SM_MOUSEWHEELPRESENT	Nonzero if a mouse with a vertical scroll wheel is installed; otherwise 0.
SM_NETWORK	The least significant bit is set if a network is present; otherwise, it is cleared.
SM_PENWINDOWS	Nonzero if the Microsoft Windows for Pen computing extensions are installed; zero otherwise.
SM_REMOTECONTROL	This system metric is used in a Terminal Services environment to determine if the current Terminal Server session is being remotely controlled. Its value is nonzero if the current session is remotely controlled; otherwise, 0.
SM_REMOTESESSION	This system metric is used in a Terminal Services environment. If the calling process is associated with a Terminal Services client session, the return value is nonzero. If the calling process is associated with the Terminal Services console session, the return value is 0.
SM_SAMEDISPLAYFORMAT	Nonzero if all the display monitors have the same color format, otherwise, 0.
SM_SECURE	This system metric should be ignored; it always returns 0.
SM_SERVERR2	The build number if the system is Windows Server 2003 R2; otherwise, 0.
SM_SHOWSOUNDS	Nonzero if the user requires an application to present information visually in situations where it would otherwise present the information only in audible form; otherwise, 0.

Continued on next page

Table 763 – continued from previous page

Value	Description
SM_SHUTTINGDOWN	Nonzero if the current session is shutting down; otherwise, 0.
SM_SLOWMACHINE	Nonzero if the computer has a low-end (slow) processor; otherwise, 0.
SM_STARTER	Nonzero if the current operating system is Windows 7 Starter Edition, Windows Vista Starter, or Windows XP Starter Edition; otherwise, 0.
SM_SWAPBUTTON	Nonzero if the meanings of the left and right mouse buttons are swapped; otherwise, 0.
SM_TABLETPC	Nonzero if the current operating system is the Windows XP Tablet PC edition or if the current operating system is Windows Vista or Windows 7 and the Tablet PC Input service is started; otherwise, 0.
SM_XVIRTUALSCREEN	The coordinates for the left side of the virtual screen.
SM_YVIRTUALSCREEN	The coordinates for the top of the virtual screen.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateSystemMetricIndexType ==

The EntityStateSystemMetricIndexType complex type defines the different values that are valid for the index entity of a systemmetric_state. These values describe the system metric or configuration setting to be retrieved. The empty string is also allowed as a valid value to support an empty element that is found when a variable reference is used within the index entity. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values. Please note that the values identified are for the index entity and are not valid values for the datatype attribute.

Restricts: oval-def:EntityStateStringType

Table 764: Enumeration Values

Value	Description
SM_ARRANGE	The flags that specify how the system arranged minimized windows.
SM_CLEANBOOT	The value that specifies how the system is started.
SM_CMONITORS	The number of display monitors on a desktop.
SM_CMOUSEBUTTONS	The number of buttons on a mouse, or zero if no mouse is installed.
SM_CXBORDER	The width of a window border, in pixels. This is equivalent to the SM_CXEDGE value for windows with the 3-D look.
SM_CXCURSOR	The width of a cursor, in pixels. The system cannot create cursors of other sizes.
SM_CXDLGFRAME	This value is the same as SM_CXFIXEDFRAME.
SM_CXDOUBLECLK	The width of the rectangle around the location of a first click in a double-click sequence, in pixels.
SM_CXDRAG	The number of pixels on either side of a mouse-down point that the mouse pointer can move before a drag operation begins.
SM_CXEDGE	The width of a 3-D border, in pixels. This metric is the 3-D counterpart of SM_CXBORDER.
SM_CXFIXEDFRAME	The thickness of the frame around the perimeter of a window that has a caption but is not sizable, in pixels.
SM_CXFOCUSBORDER	The width of the left and right edges of the focus rectangle that the DrawFocusRect draws.
SM_CXFRAME	This value is the same as SM_CXSIZEFRAME.

Continued on next page

Table 764 – continued from previous page

Value	Description
SM_CXFULLSCREEN	The width of the client area for a full-screen window on the primary display monitor, in pixels.
SM_CXHSCROLL	The width of the arrow bitmap on a horizontal scroll bar, in pixels.
SM_CXHTHUMB	The width of the thumb box in a horizontal scroll bar, in pixels.
SM_CXICON	The default width of an icon, in pixels.
SM_CXICONSPACING	The width of a grid cell for items in large icon view, in pixels.
SM_CXMAXIMIZED	The default width, in pixels, of a maximized top-level window on the primary display monitor.
SM_CXMAXTRACK	The default maximum width of a window that has a caption and sizing borders, in pixels.
SM_CXMENUCHECK	The width of the default menu check-mark bitmap, in pixels.
SM_CXMENUSIZE	The width of menu bar buttons, such as the child window close button that is used in the multiple document interface, in pixels.
SM_CXMIN	The minimum width of a window, in pixels.
SM_CXMINIMIZED	The width of a minimized window, in pixels.
SM_CXMINSPACING	The width of a grid cell for a minimized window, in pixels.
SM_CXMINTRACK	The minimum tracking width of a window, in pixels.

Continued on next page

Table 764 – continued from previous page

Value	Description
SM_CXPADDEDBORDER	The amount of border padding for captioned windows, in pixels.
SM_CXSCREEN	The width of the screen of the primary display monitor, in pixels.
SM_CXSIZE	The width of a button in a window caption or title bar, in pixels.
SM_CXSIZEFRAME	The thickness of the sizing border around the perimeter of a window that can be resized, in pixels.
SM_CXSMICON	The recommended width of a small icon, in pixels.
SM_CXSMSIZE	The width of small caption buttons, in pixels.
SM_CXVIRTUALSCREEN	The width of the virtual screen, in pixels.
SM_CXVSCROLL	The width of a vertical scroll bar, in pixels.
SM_CYBORDER	The height of a window border, in pixels.
SM_CYCAPTION	The height of a caption area, in pixels.
SM_CYCURSOR	The height of a cursor, in pixels.
SM_CYDLGFRAME	This value is the same as SM_CYFIXEDFRAME.
SM_CYDOUBLECLK	The height of the rectangle around the location of a first click in a double-click sequence, in pixels.
SM_CYDRAG	The number of pixels above and below a mouse-down point that the mouse pointer can move before a drag operation begins.

Continued on next page

Table 764 – continued from previous page

Value	Description
SM_CYEDGE	The height of a 3-D border, in pixels. This is the 3-D counterpart of SM_CYBORDER.
SM_CYFIXEDFRAME	The thickness of the frame around the perimeter of a window that has a caption but is not sizable, in pixels.
SM_CYFOCUSBORDER	The height of the top and bottom edges of the focus rectangle drawn by DrawFocusRect. This value is in pixels.
SM_CYFRAME	This value is the same as SM_CYSIZEFRAME.
SM_CYFULLSCREEN	The height of the client area for a full-screen window on the primary display monitor, in pixels.
SM_CYHSCROLL	The height of a horizontal scroll bar, in pixels.
SM_CYICON	The default height of an icon, in pixels.
SM_CYICONSPACING	The height of a grid cell for items in large icon view, in pixels.
SM_CYKANJIWINDOW	For double byte character set versions of the system, this is the height of the Kanji window at the bottom of the screen, in pixels.
SM_CYMAXIMIZED	The default height, in pixels, of a maximized top-level window on the primary display monitor.
SM_CYMAXTRACK	The default maximum height of a window that has a caption and sizing borders, in pixels.
SM_CYMENU	The height of a single-line menu bar, in pixels.
SM_CYMENUCHECK	The height of the default menu check-mark bitmap, in pixels.

Continued on next page

Table 764 – continued from previous page

Value	Description
SM_CYMENUSIZE	The height of menu bar buttons, such as the child window close button that is used in the multiple document interface, in pixels.
SM_CYMIN	The minimum height of a window, in pixels.
SM_CYMINIMIZED	The height of a minimized window, in pixels.
SM_CYMINSPACING	The height of a grid cell for a minimized window, in pixels.
SM_CYMINTRACK	The minimum tracking height of a window, in pixels.
SM_CYSCREEN	The height of the screen of the primary display monitor, in pixels.
SM_CYSIZE	The height of a button in a window caption or title bar, in pixels.
SM_CYSIZEFRAME	The thickness of the sizing border around the perimeter of a window that can be resized, in pixels.
SM_CYSMCAPTION	The height of a small caption, in pixels.
SM_CYSMICON	The recommended height of a small icon, in pixels.
SM_CYSMSIZE	The height of small caption buttons, in pixels.
SM_CYVIRTUALSCREEN	The height of the virtual screen, in pixels. The virtual screen is the bounding rectangle of all display monitors.
SM_CYVSCROLL	The height of the arrow bitmap on a vertical scroll bar, in pixels.

Continued on next page

Table 764 – continued from previous page

Value	Description
SM_CYVTHUMB	The height of the thumb box in a vertical scroll bar, in pixels.
SM_DBCSENABLED	Nonzero if User32.dll supports DBCS; otherwise, 0.
SM_DEBUG	Nonzero if the debug version of User.exe is installed; otherwise, 0.
SM_DIGITIZER	Nonzero if the current operating system is Windows 7 or Windows Server 2008 R2 and the Tablet PC Input service is started; otherwise, 0. The return value is a bitmask that specifies the type of digitizer input supported by the device.
SM_IMMENABLED	Nonzero if Input Method Manager/Input Method Editor features are enabled; otherwise, 0.
SM_MAXIMUMTOUCHES	Nonzero if there are digitizers in the system; otherwise, 0.
SM_MEDIACENTER	Nonzero if the current operating system is the Windows XP, Media Center Edition, 0 if not.
SM_MENUDROPALIGNMENT	Nonzero if drop-down menus are right-aligned with the corresponding menu-bar item; 0 if the menus are left-aligned.
SM_MIDEASTENABLED	Nonzero if the system is enabled for Hebrew and Arabic languages, 0 if not.
SM_MOUSEPRESENT	Nonzero if a mouse is installed; otherwise, 0.
SM_MOUSEHORIZONTALWHEELPRESENT	Nonzero if a mouse with a horizontal scroll wheel is installed; otherwise 0.
SM_MOUSEWHEELPRESENT	Nonzero if a mouse with a vertical scroll wheel is installed; otherwise 0.

Continued on next page

Table 764 – continued from previous page

Value	Description
SM_NETWORK	The least significant bit is set if a network is present; otherwise, it is cleared.
SM_PENWINDOWS	Nonzero if the Microsoft Windows for Pen computing extensions are installed; zero otherwise.
SM_REMOTECONTROL	This system metric is used in a Terminal Services environment to determine if the current Terminal Server session is being remotely controlled. Its value is nonzero if the current session is remotely controlled; otherwise, 0.
SM_REMOTESESSION	This system metric is used in a Terminal Services environment. If the calling process is associated with a Terminal Services client session, the return value is nonzero. If the calling process is associated with the Terminal Services console session, the return value is 0.
SM_SAMEDISPLAYFORMAT	Nonzero if all the display monitors have the same color format, otherwise, 0.
SM_SECURE	This system metric should be ignored; it always returns 0.
SM_SERVERR2	The build number if the system is Windows Server 2003 R2; otherwise, 0.
SM_SHOWSOUNDS	Nonzero if the user requires an application to present information visually in situations where it would otherwise present the information only in audible form; otherwise, 0.
SM_SHUTTINGDOWN	Nonzero if the current session is shutting down; otherwise, 0.
SM_SLOWMACHINE	Nonzero if the computer has a low-end (slow) processor; otherwise, 0.

Continued on next page

Table 764 – continued from previous page

Value	Description
SM_STARTER	Nonzero if the current operating system is Windows 7 Starter Edition, Windows Vista Starter, or Windows XP Starter Edition; otherwise, 0.
SM_SWAPBUTTON	Nonzero if the meanings of the left and right mouse buttons are swapped; otherwise, 0.
SM_TABLETPC	Nonzero if the current operating system is the Windows XP Tablet PC edition or if the current operating system is Windows Vista or Windows 7 and the Tablet PC Input service is started; otherwise, 0.
SM_XVIRTUALSCREEN	The coordinates for the left side of the virtual screen.
SM_YVIRTUALSCREEN	The coordinates for the top of the virtual screen.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityObjectGUIDType ==

The EntityObjectGUIDType restricts a string value to a representation of a GUID, used for module ID. The empty string is also allowed to support empty element associated with variable references. 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.

Restricts: oval-def:EntityObjectStringType

Pattern: ({[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{12}}){0,}

== EntityStateGUIDType ==

The EntityStateGUIDType restricts a string value to a representation of a GUID, used for module ID. The empty string is also allowed to support empty element associated with variable references. 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.

Restricts: oval-def:EntityStateStringType

Pattern: ({[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{12}}){0,}

== EntityObjectCmdletVerbType ==

The EntityObjectCmdletVerbType restricts a string value to a set of allow cmdlet verbs. The empty string is also allowed to support empty element associated with variable references. 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.

Restricts: oval-def:EntityObjectStringType

Table 765: Enumeration Values

Value	Description
Approve	The Approve verb confirms or agrees to the status of a resource or process.
Assert	The Assert verb affirms the state of a resource.
Compare	The Compare verb evaluates the data from one resource against the data from another resource.
Confirm	The Confirm verb acknowledges, verifies, or validates, the state of a resource or process.
Find	The Find verb looks for an object in a container that is unknown, implied, optional, or specified.
Get	The Get verb specifies an action that retrieves a resource.
Import	The Import verb creates a resource from data that is stored in a persistent data store (such as a file) or in an interchange format.
Measure	The Measure verb identifies resources that are consumed by a specified operation, or retrieves statistics about a resource.
Read	The Read verb acquires information from a source.
Request	The Request verb asks for a resource or asks for permissions.
Resolve	The Resolve verb maps a shorthand representation of a resource to a more complete representation.
Search	The Search verb creates a reference to a resource in a container.
Select	The Select verb locates a resource in a container.

== EntityStateCmdletVerbType ==

The EntityStateCmdletVerbType restricts a string value to a set of allow cmdlet verbs. The empty string is also allowed to support empty element associated with variable references. 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.

Restricts: oval-def:EntityStateStringType

Table 766: Enumeration Values

Value	Description
Approve	The Approve verb confirms or agrees to the status of a resource or process.
Assert	The Assert verb affirms the state of a resource.
Compare	The Compare verb evaluates the data from one resource against the data from another resource.
Confirm	The Confirm verb acknowledges, verifies, or validates, the state of a resource or process.
Find	The Find verb looks for an object in a container that is unknown, implied, optional, or specified.
Get	The Get verb specifies an action that retrieves a resource.
Import	The Import verb creates a resource from data that is stored in a persistent data store (such as a file) or in an interchange format.
Measure	The Measure verb identifies resources that are consumed by a specified operation, or retrieves statistics about a resource.
Read	The Read verb acquires information from a source.
Request	The Request verb asks for a resource or asks for permissions.
Resolve	The Resolve verb maps a shorthand representation of a resource to a more complete representation.
Search	The Search verb creates a reference to a resource in a container.
Select	The Select verb locates a resource in a container.

== EntityStateWindowsViewType ==

The EntityStateWindowsViewType restricts a string value to a specific set of values: 32-bit and 64-bit. These values describe the different values possible for the windows view behavior.

Restricts: oval-def:EntityStateStringType

Table 767: Enumeration Values

Value	Description
32_bit	Indicates the 32_bit windows view.
64_bit	Indicates the 64_bit windows view.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityObjectUserRightType ==

The EntityObjectUserRightType restricts a string value to a specific set of values that describe the different user rights/privileges. The empty string is also allowed to support empty element associated with variable references. 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.

Restricts: oval-def:EntityObjectStringType

Table 768: Enumeration Values

Value	Description
SE_ASSIGNPRIMARYTOKEN_NAME	This privilege is required to assign the primary token of a process.
SE_AUDIT_NAME	This privilege is required to generate audit-log entries.
SE_BACKUP_NAME	This privilege is required to perform backup operations.
SE_CHANGE_NOTIFY_NAME	This privilege is required to receive notifications of changes to files or directories.

Continued on next page

Table 768 – continued from previous page

Value	Description
SE_CREATE_GLOBAL_NAME	This privilege is required to create named file mapping objects in the global namespace during Terminal Services sessions.
SE_CREATE_PAGEFILE_NAME	This privilege is required to create a paging file.
SE_CREATE_PERMANENT_NAME	This privilege is required to create a permanent object.
SE_CREATE_SYMBOLIC_LINK_NAME	This privilege is required to create a symbolic link.
SE_CREATE_TOKEN_NAME	This privilege is required to create a primary token.
SE_DEBUG_NAME	This privilege is required to debug and adjust the memory of a process owned by another account.
SE_ENABLE_DELEGATION_NAME	This privilege is required to mark user and computer accounts as trusted for delegation.
SE_IMPERSONATE_NAME	This privilege is required to impersonate.
SE_INC_BASE_PRIORITY_NAME	This privilege is required to increase the base priority of a process.
SE_INCREASE_QUOTA_NAME	This privilege is required to increase the quota assigned to a process.
SE_INC_WORKING_SET_NAME	This privilege is required to allocate more memory for applications that run in the context of users.
SE_LOAD_DRIVER_NAME	This privilege is required to load or unload a device driver.
SE_LOCK_MEMORY_NAME	This privilege is required to lock physical pages in memory.

Continued on next page

Table 768 – continued from previous page

Value	Description
SE_MACHINE_ACCOUNT_NAME	This privilege is required to create a computer account.
SE_MANAGE_VOLUME_NAME	This privilege is required to enable volume management privileges.
SE_PROF_SINGLE_PROCESS_NAME	This privilege is required to gather profiling information for a single process.
SE_RELABEL_NAME	This privilege is required to modify the mandatory integrity level of an object.
SE_REMOTE_SHUTDOWN_NAME	This privilege is required to shut down a system using a network request.
SE_RESTORE_NAME	This privilege is required to perform restore operations.
SE_SECURITY_NAME	This privilege is required to perform a number of security-related functions, such as controlling and viewing audit messages.
SE_SHUTDOWN_NAME	This privilege is required to shut down a local system.
SE_SYNC_AGENT_NAME	This privilege is required for a domain controller to use the Lightweight Directory Access Protocol directory synchronization services.
SE_SYSTEM_ENVIRONMENT_NAME	This privilege is required to modify the nonvolatile RAM of systems that use this type of memory to store configuration information.
SE_SYSTEM_PROFILE_NAME	This privilege is required to gather profiling information for the entire system.
SE_SYSTEMTIME_NAME	This privilege is required to modify the system time.

Continued on next page

Table 768 – continued from previous page

Value	Description
SE_TAKE_OWNERSHIP_NAME	This privilege is required to take ownership of an object without being granted discretionary access.
SE_TCB_NAME	This privilege identifies its holder as part of the trusted computer base.
SE_TIME_ZONE_NAME	This privilege is required to adjust the time zone associated with the computer's internal clock.
SE_TRUSTED_CREDMAN_ACCESS_NAME	This privilege is required to access Credential Manager as a trusted caller.
SE_UNDOCK_NAME	This privilege is required to undock a laptop.
SE_UNSOLICITED_INPUT_NAME	This privilege is required to read unsolicited input from a terminal device.
SE_BATCH_LOGON_NAME	This account right is required for an account to log on using the batch logon type.
SE_DENY_BATCH_LOGON_NAME	This account right explicitly denies an account the right to log on using the batch logon type.
SE_DENY_INTERACTIVE_LOGON_NAME	This account right explicitly denies an account the right to log on using the interactive logon type.
SE_DENY_NETWORK_LOGON_NAME	This account right explicitly denies an account the right to log on using the network logon type.
SE_DENY_REMOTE_INTERACTIVE_LOGON_NAME	This account right explicitly denies an account the right to log on remotely using the interactive logon type.
SE_DENY_SERVICE_LOGON_NAME	This account right explicitly denies an account the right to log on using the service logon type.

Continued on next page

Table 768 – continued from previous page

Value	Description
SE_INTERACTIVE_LOGON_NAME	This account right is required for an account to log on using the interactive logon type.
SE_NETWORK_LOGON_NAME	This account right is required for an account to log on using the network logon type.
SE_REMOTE_INTERACTIVE_LOGON_NAME	This account right is required for an account to log on remotely using the interactive logon type.
SE_SERVICE_LOGON_NAME	This account right is required for an account to log on using the service logon type.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateUserRightType ==

The EntityStateUserRightType restricts a string value to a specific set of values that describe the different user rights/privileges. The empty string is also allowed to support empty element associated with variable references. 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.

Restricts: oval-def:EntityStateStringType

Table 769: Enumeration Values

Value	Description
SE_ASSIGNPRIMARYTOKEN_NAME	This privilege is required to assign the primary token of a process.
SE_AUDIT_NAME	This privilege is required to generate audit-log entries.
SE_BACKUP_NAME	This privilege is required to perform backup operations.
SE_CHANGE_NOTIFY_NAME	This privilege is required to receive notifications of changes to files or directories.

Continued on next page

Table 769 – continued from previous page

Value	Description
SE_CREATE_GLOBAL_NAME	This privilege is required to create named file mapping objects in the global namespace during Terminal Services sessions.
SE_CREATE_PAGEFILE_NAME	This privilege is required to create a paging file.
SE_CREATE_PERMANENT_NAME	This privilege is required to create a permanent object.
SE_CREATE_SYMBOLIC_LINK_NAME	This privilege is required to create a symbolic link.
SE_CREATE_TOKEN_NAME	This privilege is required to create a primary token.
SE_DEBUG_NAME	This privilege is required to debug and adjust the memory of a process owned by another account.
SE_ENABLE_DELEGATION_NAME	This privilege is required to mark user and computer accounts as trusted for delegation.
SE_IMPERSONATE_NAME	This privilege is required to impersonate.
SE_INC_BASE_PRIORITY_NAME	This privilege is required to increase the base priority of a process.
SE_INCREASE_QUOTA_NAME	This privilege is required to increase the quota assigned to a process.
SE_INC_WORKING_SET_NAME	This privilege is required to allocate more memory for applications that run in the context of users.
SE_LOAD_DRIVER_NAME	This privilege is required to load or unload a device driver.
SE_LOCK_MEMORY_NAME	This privilege is required to lock physical pages in memory.

Continued on next page

Table 769 – continued from previous page

Value	Description
SE_MACHINE_ACCOUNT_NAME	This privilege is required to create a computer account.
SE_MANAGE_VOLUME_NAME	This privilege is required to enable volume management privileges.
SE_PROF_SINGLE_PROCESS_NAME	This privilege is required to gather profiling information for a single process.
SE_RELABEL_NAME	This privilege is required to modify the mandatory integrity level of an object.
SE_REMOTE_SHUTDOWN_NAME	This privilege is required to shut down a system using a network request.
SE_RESTORE_NAME	This privilege is required to perform restore operations.
SE_SECURITY_NAME	This privilege is required to perform a number of security-related functions, such as controlling and viewing audit messages.
SE_SHUTDOWN_NAME	This privilege is required to shut down a local system.
SE_SYNC_AGENT_NAME	This privilege is required for a domain controller to use the Lightweight Directory Access Protocol directory synchronization services.
SE_SYSTEM_ENVIRONMENT_NAME	This privilege is required to modify the nonvolatile RAM of systems that use this type of memory to store configuration information.
SE_SYSTEM_PROFILE_NAME	This privilege is required to gather profiling information for the entire system.
SE_SYSTEMTIME_NAME	This privilege is required to modify the system time.

Continued on next page

Table 769 – continued from previous page

Value	Description
SE_TAKE_OWNERSHIP_NAME	This privilege is required to take ownership of an object without being granted discretionary access.
SE_TCB_NAME	This privilege identifies its holder as part of the trusted computer base.
SE_TIME_ZONE_NAME	This privilege is required to adjust the time zone associated with the computer's internal clock.
SE_TRUSTED_CREDMAN_ACCESS_NAME	This privilege is required to access Credential Manager as a trusted caller.
SE_UNDOCK_NAME	This privilege is required to undock a laptop.
SE_UNSOLICITED_INPUT_NAME	This privilege is required to read unsolicited input from a terminal device.
SE_BATCH_LOGON_NAME	This account right is required for an account to log on using the batch logon type.
SE_DENY_BATCH_LOGON_NAME	This account right explicitly denies an account the right to log on using the batch logon type.
SE_DENY_INTERACTIVE_LOGON_NAME	This account right explicitly denies an account the right to log on using the interactive logon type.
SE_DENY_NETWORK_LOGON_NAME	This account right explicitly denies an account the right to log on using the network logon type.
SE_DENY_REMOTE_INTERACTIVE_LOGON_NAME	This account right explicitly denies an account the right to log on remotely using the interactive logon type.
SE_DENY_SERVICE_LOGON_NAME	This account right explicitly denies an account the right to log on using the service logon type.

Continued on next page

Table 769 – continued from previous page

Value	Description
SE_INTERACTIVE_LOGON_NAME	This account right is required for an account to log on using the interactive logon type.
SE_NETWORK_LOGON_NAME	This account right is required for an account to log on using the network logon type.
SE_REMOTE_INTERACTIVE_LOGON_NAME	This account right is required for an account to log on remotely using the interactive logon type.
SE_SERVICE_LOGON_NAME	This account right is required for an account to log on using the service logon type.
	The empty string value is permitted here to allow for empty elements associated with variable references.

Open Vulnerability and Assessment Language: Windows System Characteristics

- Schema: Windows System Characteristics
- Version: 5.11.1:1.4
- Release Date: 01/09/2017 10:00:00 PM

The following is a description of the elements, types, and attributes that compose the Windows specific system characteristic items found in Open Vulnerability and Assessment Language (OVAL). Each item is an extension of the standard item element defined in the Core System Characteristic Schema. Through extension, each item inherits a set of elements and attributes that are shared amongst all OVAL Items. Each item is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core System Characteristic Schema is not outlined here.

The OVAL Schema is maintained by the 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.cisecurity.org.

Item Listing

- *< accesstoken_item > (Deprecated)*
- *< activedirectory_item > (Deprecated)*
- *< activedirectory57_item >*
- *< auditeventpolicy_item >*
- *< auditeventpolicysubcategories_item >*
- *< cmdlet_item >*
- *< dnscache_item >*

- *< file_item >*
- *< fileauditedpermissions_item >*
- *< fileeffectiverights_item >*
- *< group_item > (Deprecated)*
- *< group_sid_item >*
- *< interface_item >*
- *< junction_item >*
- *< license_item >*
- *< lockoutpolicy_item >*
- *< metabase_item >*
- *< ntuser_item >*
- *< passwordpolicy_item >*
- *< peheader_item >*
- *< port_item >*
- *< printereffectiverights_item >*
- *< process_item >*
- *< registry_item >*
- *< regkeyauditedpermissions_item >*
- *< regkeyeffectiverights_item >*
- *< service_item >*
- *< serviceeffectiverights_item >*
- *< sharedresource_item >*
- *< sharedresourceauditedpermissions_item >*
- *< sharedresourceeffectiverights_item >*
- *< sid_item >*
- *< sid_sid_item >*
- *< systemmetric_item >*
- *< uac_item >*
- *< user_item > (Deprecated)*
- *< user_sid_item >*
- *< userright_item >*
- *< volume_item >*
- *< wmi_item > (Deprecated)*
- *< wmi57_item >*
- *< wuaupdatesearcher_item >*

< accesstoken_item > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.11
- Reason: Replaced by the userright_item. The accesstoken_test suffers from scalability issues when run on a domain controller and should not be used. See the userright_item.
- Comment: This object has been deprecated and may be removed in a future version of the language.

The access token item holds information about the individual privileges and rights associated with a specific access token. It is important to note that these privileges are specific to certain versions of Windows. As a result, the documentation for that version of Windows should be consulted for more information. Each privilege and right in the data section accepts a boolean value signifying whether the privilege is granted or not. It extends the standard ItemType as defined in the oval-system-characteristics schema and one should refer to the ItemType description for more information.

Extends: oval-sc:ItemType

Child Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
security_principle	oval-sc:EntityItemStringType (0..1)	Security principles include users or groups with either loc
seassignprimarytokenprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows a parent process to re
seauditprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows a process to generate
sebackupprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows the user to circumver
sechangenotifyprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows the user to pass throu
secreateglobalprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows the user to create nan
secreatepagefileprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows the user to create and
secreatepermanentprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows a process to create a
secreatesymboliclinkprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows a user create a symbo
secreatetokenprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows a process to create an
sedebugprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows the user to attach a de
seenabledelegationprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows the user to change the
seimpersonateprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows the user to imperson
seincreasebasepriorityprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows a user to increase the
seincreasequotaprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows a process that has acc
seincreaseworkingsetprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows a user to increase a p
seloaddriverprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows a user to install and r
selockmemoryprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows a process to keep dat
semachineaccountprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows the user to add a com
semanagevolumeprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows a non-administrative
seprofilesingleprocessprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows a user to sample the p
serelabelprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows a user to modify an o
seremoteshutdownprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows a user to shut down a
serestoreprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows a user to circumvent
sesecurityprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows a user to specify obje
seshutdownprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows a user to shut down t
sesyncagentprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows a process to read all

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
sesystemenvironmentprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows modification of syste
sesystemprofileprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows a user to sample the p
sesystemtimeprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows the user to adjust the
setakeownershipprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows a user to take owners
setcbprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows a process to assume t
setimezoneprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows a user to change the t
seundockprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows the user of a portable
seunsolicitedinputprivilege	oval-sc:EntityItemBoolType (0..1)	If this privilege is enabled, it allows the user to read unsol
sebatchlogonright	oval-sc:EntityItemBoolType (0..1)	If an account is assigned this right, it can log on using the
seinteractivelogonright	oval-sc:EntityItemBoolType (0..1)	If an account is assigned this right, it can log on using the
senetworklogonright	oval-sc:EntityItemBoolType (0..1)	If an account is assigned this right, it can log on using the
seremoteinteractivelogonright	oval-sc:EntityItemBoolType (0..1)	If an account is assigned this right, it can log on to the co
seservicelogonright	oval-sc:EntityItemBoolType (0..1)	If an account is assigned this right, it can log on using the
sedenybatchLogonright	oval-sc:EntityItemBoolType (0..1)	If an account is assigned this right, it is explicitly denied t
sedenyinteractivelogonright	oval-sc:EntityItemBoolType (0..1)	If an account is assigned this right, it is explicitly denied t
sedenynetworklogonright	oval-sc:EntityItemBoolType (0..1)	If an account is assigned this right, it is explicitly denied t
sedenyremoteInteractivelogonright	oval-sc:EntityItemBoolType (0..1)	If an account is assigned this right, it is explicitly denied t
sedenyservicelogonright	oval-sc:EntityItemBoolType (0..1)	If an account is assigned this right, it is explicitly denied t
setrustedcredmanaccessnameright	oval-sc:EntityItemBoolType (0..1)	If an account is assigned this right, it can access the Cred

< activedirectory_item > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.11.1:1.2
- Reason: Use the original activedirectory_item. The activedirectory57_test suffers from ambiguity; it was never adequately specified, and it does not even seem possible to have structured data in the context of the enumerated AdstypeTypes. Use the original activedirectory_test instead.
- Comment: This object has been deprecated and may be removed in a future version of the language.

The active directory item holds information about specific entries in the Windows Active Directory. It extends the standard ItemType as defined in the oval-system-characteristics schema and one should refer to the ItemType description for more information.

Note that this ite supports only simple (string based) value collection. For more complex values see the activedirectory57_item.

Extends: oval-sc:ItemType

Child Elements

Table 771: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
naming_context	win-sc:EntityItemNamingContextType (0..1)	Each object in active directory exists under a certain naming context (also known as a partition). A naming context is defined as a single object in the Directory Information Tree (DIT) along with every object in the tree subordinate to it. There are three default naming contexts in Active Directory: domain, configuration, and schema.
relative_dn	oval-sc:EntityItemStringType (0..1)	The relative_dn field is used to uniquely identify an object inside the specified naming context. It contains all the parts of the objects distinguished name except those outlined by the naming context. If the xsi:nil attribute is set to true, then the item being represented is the higher level naming context.
attribute	oval-sc:EntityItemStringType (0..1)	Specifies a named value contained by the object.
object_class	oval-sc:EntityItemStringType (0..1)	The name of the class of which the object is an instance.
adstype	win-sc:EntityItemAdstypeType (0..1)	Specifies the type of information that the specified attribute represents.
value	oval-sc:EntityItemAnySimpleType (0..unbounded)	The actual value of the specified active directory attribute.

< activedirectory57_item >

The activedirectory57_item holds information about specific entries in the Windows Active Directory. It extends the standard ItemType as defined in the oval-system-characteristics schema and one should refer to the ItemType description for more information.

Note that this item supports complex values that are in the form of a record. For simple (string based) value collection see the activedirectory_item.

Extends: oval-sc:ItemType

Child Elements

Table 772: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
naming_context	win-sc:EntityItemNamingContextType (0..1)	Each object in active directory exists under a certain naming context (also known as a partition). A naming context is defined as a single object in the Directory Information Tree (DIT) along with every object in the tree subordinate to it. There are three default naming contexts in Active Directory: domain, configuration, and schema.
relative_dn	oval-sc:EntityItemStringType (0..1)	The relative_dn field is used to uniquely identify an object inside the specified naming context. It contains all the parts of the objects distinguished name except those outlined by the naming context. If the xsi:nil attribute is set to true, then the item being represented is the higher level naming context.
attribute	oval-sc:EntityItemStringType (0..1)	Specifies a named value contained by the object.
object_class	oval-sc:EntityItemStringType (0..1)	The name of the class of which the object is an instance.
adstype	win-sc:EntityItemAdstypeType (0..1)	Specifies the type of information that the specified attribute represents.
value	oval-sc:EntityItemRecordType (0..unbounded)	The actual value of the specified Active Directory attribute. Note that while an Active Directory attribute can contain structured data where it is necessary to collect multiple related fields that can be described by the 'record' datatype, it is not always the case. It also is possible that an Active Directory attribute can contain only a single value or an array of values. In these cases, there is not a name to uniquely identify the corresponding field(s) which is a requirement for fields in the 'record' datatype. As a result, the name of the Active Directory attribute will be used to uniquely identify the field(s) and satisfy this requirement. If the Active Directory attribute contains a single value, the 'record' will have a single field identified by the name of the Active Directory attribute. If the Active Directory attribute contains an array of values, the 'record' will have multiple fields all identified by the name of the Active Directory attribute

< auditeventpolicy_item >

The auditeventpolicy item enumerates the different types of events the system should audit. The defined values are found in window's POLICY_AUDIT_EVENT_TYPE enumeration and accessed through the LsaQueryInformationPolicy when the InformationClass parameters are set to PolicyAuditEventsInformation. It extends the standard ItemType as defined in the oval-system-characteristics schema and one should refer to the ItemType description for more information.

Note that when audinting is disabled each of the entities listed below should be set to 'AUDIT_NONE'.

Extends: oval-sc:ItemType

Child Elements

Table 773: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
account_logon	win-sc:EntityItemAuditType (0..1)	Audit attempts to log on to or log off of the system. Also, audit attempts to make a network connection.
account_management	win-sc:EntityItemAuditType (0..1)	Audit attempts to create, delete, or change user or group accounts. Also, audit password changes.
detailed_tracking	win-sc:EntityItemAuditType (0..1)	Audit specific events, such as program activation, some forms of handle duplication, indirect access to an object, and process exit.
directory_service_access	win-sc:EntityItemAuditType (0..1)	Audit attempts to access the directory service.
logon	win-sc:EntityItemAuditType (0..1)	Audit attempts to log on to or log off of the system. Also, audit attempts to make a network connection.
object_access	win-sc:EntityItemAuditType (0..1)	Audit attempts to access securable objects, such as files.
policy_change	win-sc:EntityItemAuditType (0..1)	Audit attempts to change Policy object rules.
privilege_use	win-sc:EntityItemAuditType (0..1)	Audit attempts to use privileges.
system	win-sc:EntityItemAuditType (0..1)	Audit attempts to shut down or restart the computer. Also, audit events that affect system security or the security log.

< auditeventpolicysubcategories_item >

The auditeventpolicysubcategories_item is used to hold information about the audit event policy settings on a Windows system. These settings are used to specify which system and network events are monitored. For example, if the credential_validation element has a value of AUDIT_FAILURE, it means that the system is configured to log all unsuccessful attempts to validate a user account on a system. It is important to note that these audit event policy settings are specific to certain versions of Windows. As a result, the documentation for that version of Windows should be consulted for more information on each setting. It extends the standard ItemType as defined in the oval-system-characteristics schema and one should refer to the ItemType description for more information.

Note that when audinting is disabled each of the entities listed below should be set to 'AUDIT_NONE'.

Extends: oval-sc:ItemType

Child Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
credential_validation	win-sc:EntityItemAuditType (0..1)	Audit the events produced during the validation of
kerberos_authentication_service	win-sc:EntityItemAuditType (0..1)	Audit the events produced by Kerberos authenticat
kerberos_service_ticket_operations	win-sc:EntityItemAuditType (0..1)	Audit the events produced by Kerberos service tick
kerberos_ticket_events (Deprecated)	win-sc:EntityItemAuditType (0..1)	Audit the events produced during the validation of
other_account_logon_events	win-sc:EntityItemAuditType (0..1)	Audit the events produced by changes to user accou
application_group_management	win-sc:EntityItemAuditType (0..1)	Audit the events produced by changes to applicatio
computer_account_management	win-sc:EntityItemAuditType (0..1)	Audit the events produced by changes to computer
distribution_group_management	win-sc:EntityItemAuditType (0..1)	Audit the events produced by changes to distributio
other_account_management_events	win-sc:EntityItemAuditType (0..1)	Audit the events produced by other user account ch
security_group_management	win-sc:EntityItemAuditType (0..1)	Audit the events produced by changes to security g
user_account_management	win-sc:EntityItemAuditType (0..1)	Audit the events produced by changes to user accou
dpapi_activity	win-sc:EntityItemAuditType (0..1)	Audit the events produced when requests are made
process_creation	win-sc:EntityItemAuditType (0..1)	Audit the events produced when a process is create
process_termination	win-sc:EntityItemAuditType (0..1)	Audit the events produced when a process ends. Th
rpc_events	win-sc:EntityItemAuditType (0..1)	Audit the events produced by inbound remote proc
directory_service_access	win-sc:EntityItemAuditType (0..1)	Audit the events produced when a Active Directory
directory_service_changes	win-sc:EntityItemAuditType (0..1)	Audit the events produced when changes are made
directory_service_replication	win-sc:EntityItemAuditType (0..1)	Audit the events produced when two Active Directo
detailed_directory_service_replication	win-sc:EntityItemAuditType (0..1)	Audit the events produced by detailed Active Direc
account_lockout	win-sc:EntityItemAuditType (0..1)	Audit the events produced by a failed attempt to lo
ipsec_extended_mode	win-sc:EntityItemAuditType (0..1)	Audit the events produced by Internet Key Exchan
ipsec_main_mode	win-sc:EntityItemAuditType (0..1)	Audit the events produced by Internet Key Exchan
ipsec_quick_mode	win-sc:EntityItemAuditType (0..1)	Audit the events produced by Internet Key Exchan
logoff	win-sc:EntityItemAuditType (0..1)	Audit the events produced by closing a logon sessi
logon	win-sc:EntityItemAuditType (0..1)	Audit the events produced by attempts to log onto a
network_policy_server	win-sc:EntityItemAuditType (0..1)	Audit the events produced by RADIUS and Netwo
other_logon_logoff_events	win-sc:EntityItemAuditType (0..1)	Audit the events produced by other logon/logoff ba
special_logon	win-sc:EntityItemAuditType (0..1)	Audit the events produced by special logons. This
logon_claims	win-sc:EntityItemAuditType (0..1)	Audit user and device claims information in the use
application_generated	win-sc:EntityItemAuditType (0..1)	Audit the events produced by applications that use
certification_services	win-sc:EntityItemAuditType (0..1)	Audit the events produced by operations on Active
detailed_file_share	win-sc:EntityItemAuditType (0..1)	Audit the events produced by attempts to access fil
file_share	win-sc:EntityItemAuditType (0..1)	Audit the events produced by attempts to access a s
file_system	win-sc:EntityItemAuditType (0..1)	Audit the events produced user attempts to access f
filtering_platform_connection	win-sc:EntityItemAuditType (0..1)	Audit the events produced by connections that are a
filtering_platform_packet_drop	win-sc:EntityItemAuditType (0..1)	Audit the events produced by packets that are drop
handle_manipulation	win-sc:EntityItemAuditType (0..1)	Audit the events produced when a handle is opened
kernel_object	win-sc:EntityItemAuditType (0..1)	Audit the events produced by attempts to access the
other_object_access_events	win-sc:EntityItemAuditType (0..1)	Audit the events produced by the management of T
registry	win-sc:EntityItemAuditType (0..1)	Audit the events produced by attempts to access reg
sam	win-sc:EntityItemAuditType (0..1)	Audit the events produced by attempts to access Se
removable_storage	win-sc:EntityItemAuditType (0..1)	Audit events that indicate file object access attemps
central_access_policy_staging	win-sc:EntityItemAuditType (0..1)	Audit events that indicate permission granted or de
audit_policy_change	win-sc:EntityItemAuditType (0..1)	Audit the events produced by changes in security a
authentication_policy_change	win-sc:EntityItemAuditType (0..1)	Audit the events produced by changes to the authen
authorization_policy_change	win-sc:EntityItemAuditType (0..1)	Audit the events produced by changes to the author
filtering_platform_policy_change	win-sc:EntityItemAuditType (0..1)	Audit the events produced by changes to the Windo
mpssvc_rule_level_policy_change	win-sc:EntityItemAuditType (0..1)	Audit the events produced by changes to policy rul
other_policy_change_events	win-sc:EntityItemAuditType (0..1)	Audit the events produced by other security policy

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
non_sensitive_privilege_use	win-sc:EntityItemAuditType (0..1)	Audit the events produced by the use of non-sensit
other_privilege_use_events	win-sc:EntityItemAuditType (0..1)	This is currently not used and has been reserved by
sensitive_privilege_use	win-sc:EntityItemAuditType (0..1)	Audit the events produced by the use of sensitive p
ipsec_driver	win-sc:EntityItemAuditType (0..1)	Audit the events produced by the IPsec filter driver
other_system_events	win-sc:EntityItemAuditType (0..1)	Audit the events produced by the startup and shutd
security_state_change	win-sc:EntityItemAuditType (0..1)	Audit the events produced by changes in the securi
security_system_extension	win-sc:EntityItemAuditType (0..1)	Audit the events produced by the security system e
system_integrity	win-sc:EntityItemAuditType (0..1)	Audit the events that indicate that the integrity secu
group_membership	win-sc:EntityItemAuditType (0..1)	This subcategory audits the group membership of a
pnp_activity	win-sc:EntityItemAuditType (0..1)	This subcategory audits events generated by plug a
user_device_claims	win-sc:EntityItemAuditType (0..1)	This subcategory audits the user and device claims
audit_detailedtracking_tokenrightadjusted	win-sc:EntityItemAuditType (0..1)	This subcategory audits when token privileges are

< cmdlet_item >

The cmdlet_item represents a PowerShell cmdlet, the parameters supplied to it, and the value it returned.

Extends: oval-sc:ItemType

Child Elements

Table 775: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
module_name	oval-sc:EntityItemStringType (0..1)	The name of the module that contains the cmdlet.
module_id	win-sc:EntityItemGUIDType (0..1)	The globally unique identifier for the module.
module_version	oval-sc:EntityItemVersionType (0..1)	The version of the module that contains the cmdlet in the form of MAJOR.MINOR.
verb	win-sc:EntityItemCmdletVerbType (0..1)	The cmdlet verb.
noun	oval-sc:EntityItemStringType (0..1)	The cmdlet noun.
parameters	oval-sc:EntityItemRecordType (0..1)	A list of properties (name and value pairs) as input to invoke the cmdlet.
select	oval-sc:EntityItemRecordType (0..1)	A list of fields (name and value pairs) used as input to the Select-Object cmdlet to select specific output properties.
value	oval-sc:EntityItemRecordType (0..unbounded)	The expected value represented as a set of fields (name and value pairs).

< dnscache_item >

The dnscache_item stores information retrieved from the DNS cache about a domain name, its time to live, and its corresponding IP addresses.

Extends: oval-sc:ItemType

Child Elements

Table 776: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
domain_name	oval-sc:EntityItemStringType (0..1)	The domain_name element contains a string that represents a domain name that was collected from the DNS cache on the local system.
ttl	oval-sc:EntityItemIntType (0..1)	The ttl element contains an integer that represents the time to live in seconds of the DNS cache entry.
ip_address	oval-sc:EntityItemIPAddressStringType (0..unbounded)	The ip_address element contains a string that represents an IP address associated with the specified domain name. Note that the IP address can be IPv4 or IPv6.

< file_item >

This element describes file metadata. The time information can be retrieved by the _stst function. Development_class and other version information (company, internal name, language, original_filename, product_name, product_version) can be retrieved using the VerQueryValue function.

Extends: oval-sc:ItemType

Child Elements

Table 777: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-sc:EntityItemStringType (0..1)	The filepath element specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath.
path	oval-sc:EntityItemStringType (0..1)	Specifies the directory component of the absolute path to a file on the machine.
filename	oval-sc:EntityItemStringType (0..1)	The name of the file. If the xsi:nil attribute is set to true, then the item being represented is the higher directory represented by the path entity. The other items associated with this item would then reflect the values associated with the directory.
owner	oval-sc:EntityItemStringType (0..1)	A string that contains the name of the owner. The name should be specified in the DOMAIN\username format.
size	oval-sc:EntityItemIntType (0..1)	Size of the file in bytes.
a_time	oval-sc:EntityItemIntType (0..1)	Time of last access of file. Valid on NTFS but not on FAT formatted disk drives. The string should represent the FILETIME structure which is a 64-bit value representing the number of 100-nanosecond intervals since January 1, 1601 (UTC).
c_time	oval-sc:EntityItemIntType (0..1)	Time of creation of file. Valid on NTFS but not on FAT formatted disk drives. The string should represent the FILETIME structure which is a 64-bit value representing the number of 100-nanosecond intervals since January 1, 1601 (UTC).
m_time	oval-sc:EntityItemIntType (0..1)	Time of last modification of file. The string should represent the FILETIME structure which is a 64-bit value representing the number of 100-nanosecond intervals since January 1, 1601 (UTC).
ms_checksum	oval-sc:EntityItemStringType (0..1)	The checksum of the file as supplied by Microsoft's MapFileAndCheckSum function.
version	oval-sc:EntityItemVersionType (0..1)	The version of the file.
type	win-sc:EntityItemFileTypeType (0..1)	The type child element marks whether the file item describes a named pipe, standard file, etc. These types are the return values for GetFileType. For directories, this element must have a status of 'does not exist'.
attribute	win-sc:EntityItemFileAttributeType (0..unbounded)	The attribute child elements denote the Windows file attributes associated with the file. These types are the return values for GetFileAttributes.
development_class	oval-sc:EntityItemStringType (0..1)	The development_class element allows the distinction to be made between the GDR development environment and the QFE development environment. This field holds the text found in front of the mmmmmm-nnnn version, for example srv03_gdr.
company	oval-sc:EntityItemStringType (0..1)	This entity defines the company name held within the version-information structure.
internal_name	oval-sc:EntityItemStringType (0..1)	This entity defines the internal name held within the version-information structure.
language	oval-sc:EntityItemStringType (0..1)	This entity defines the language held within the version-information structure.
orig-i-	oval-sc:EntityItemStringType	This entity defines the original filename held within the version-information structure.

< fileauditedpermissions_item >

This item stores the audited access rights of a file that a system access control list (SACL) structure grants to a specified trustee. The trustee's audited access rights are determined checking all access control entries (ACEs) in the SACL. For help with this test see the GetAuditedPermissionsFromAcl() api.

Extends: oval-sc:ItemType

Child Elements

Table 778: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-sc:EntityItemStringType (0..1)	Specifies the absolute path to a file on the machine from which the DACL was retrieved. A directory cannot be specified as a filepath.
path	oval-sc:EntityItemStringType (0..1)	This element specifies the directory component of the absolute path to a file on the machine from which the DACL was retrieved.
filename	oval-sc:EntityItemStringType (0..1)	The name of the file. If the xsi:nil attribute is set to true, then the item being represented is the higher directory represented by the path entity. The other items associated with this item would then reflect the values associated with the directory.
trustee_sid	oval-sc:EntityItemStringType (0..1)	The trustee_sid entity specifies the SID that associated a user, group, system, or program (such as a Windows service).
trustee_name (Deprecated)	oval-sc:EntityItemStringType (0..1)	This element specifies the trustee name associated with this particular SACL. A trustee can be a user, group, or program (such as a Windows service). In Windows, trustee names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, trustee names should be identified in the form: “domaintrustee name”. For local trustee names use: “computer nametrustee name”. For built-in accounts on the system, use the trustee name without a domain.
standard_delete	win-sc:EntityItemAuditType (0..1)	The right to delete the object.
standard_read_control	win-sc:EntityItemAuditType (0..1)	The right to read the information in the object’s security descriptor, not including the information in the SACL.
standard_write_dac	win-sc:EntityItemAuditType (0..1)	The right to modify the DACL in the object’s security descriptor.
standard_write_owner	win-sc:EntityItemAuditType (0..1)	The right to change the owner in the object’s security descriptor.
standard_synchronize	win-sc:EntityItemAuditType (0..1)	The right to use the object for synchronization. This enables a thread to wait until the object is in the signaled state. Some object types do not support this access right.
access_system_security	win-sc:EntityItemAuditType (0..1)	Indicates access to a system access control list (SACL).
generic_read	win-sc:EntityItemAuditType (0..1)	Read access.
generic_write	win-sc:EntityItemAuditType (0..1)	Write access.
generic_execute	win-sc:EntityItemAuditType (0..1)	Execute access.
generic_all	win-sc:EntityItemAuditType (0..1)	Read, write, and execute access.
file_read_data	win-sc:EntityItemAuditType (0..1)	Grants the right to read data from the file.
file_write_data	win-sc:EntityItemAuditType	Grants the right to write data to the file.

< fileeffectiverights_item >

This item stores the effective rights of a file that a discretionary access control list (DACL) structure grants to a specified trustee. The trustee's effective rights are determined checking all access-allowed and access-denied access control entries (ACEs) in the DACL. For help with this test see the GetEffectiveRightsFromAcl() api.

Extends: oval-sc:ItemType

Child Elements

Table 779: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-sc:EntityItemStringType (0..1)	Specifies the absolute path to a file on the machine from which the DACL was retrieved. A directory cannot be specified as a filepath.
path	oval-sc:EntityItemStringType (0..1)	This element specifies the absolute path to a file on the machine from which the DACL was retrieved.
filename	oval-sc:EntityItemStringType (0..1)	The name of the file. If the xsi:nil attribute is set to true, then the item being represented is the higher directory represented by the path entity. The other items associated with this item would then reflect the values associated with the directory.
trustee_sid	oval-sc:EntityItemStringType (0..1)	The trustee_sid entity specifies the SID that associated a user, group, system, or program (such as a Windows service).
trustee_name (Deprecated)	oval-sc:EntityItemStringType (0..1)	This element specifies the trustee name associated with this particular DACL. A trustee can be a user, group, or program (such as a Windows service). In Windows, trustee names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, trustee names should be identified in the form: “domaintrustee name”. For local trustee names use: “computer nametrustee name”. For built-in accounts on the system, use the trustee name without a domain.
standard_delete	oval-sc:EntityItemBoolType (0..1)	The right to delete the object.
standard_read_control	oval-sc:EntityItemBoolType (0..1)	The right to read the information in the object’s security descriptor, not including the information in the SACL.
standard_write_dac	oval-sc:EntityItemBoolType (0..1)	The right to modify the DACL in the object’s security descriptor.
standard_write_owner	oval-sc:EntityItemBoolType (0..1)	The right to change the owner in the object’s security descriptor.
standard_synchronize	oval-sc:EntityItemBoolType (0..1)	The right to use the object for synchronization. This enables a thread to wait until the object is in the signaled state. Some object types do not support this access right.
access_system_security	oval-sc:EntityItemBoolType (0..1)	Indicates access to a system access control list (SACL).
generic_read	oval-sc:EntityItemBoolType (0..1)	Read access.
generic_write	oval-sc:EntityItemBoolType (0..1)	Write access.
generic_execute	oval-sc:EntityItemBoolType (0..1)	Execute access.
generic_all	oval-sc:EntityItemBoolType (0..1)	Read, write, and execute access.
file_read_data	oval-sc:EntityItemBoolType (0..1)	Grants the right to read data from the file
file_write_data	oval-sc:EntityItemBoolType	Grants the right to write data to the file.

< group_item > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.11
- Reason: Replaced by the group_sid_item. This item uses trustee names for identifying accounts on the system. Trustee names are not unique and the group_sid_item, which uses trustee SIDs which are unique, should be used instead. See the group_sid_item.
- Comment: This object has been deprecated and may be removed in a future version of the language.

The Windows group_item allows the different users and subgroups, that directly belong to specific groups (identified by name), to be collected. The collected subgroups will not be resolved to find indirect user or subgroup members. If the subgroups need to be resolved, it should be done using the sid_object. Note that the user and subgroup elements can appear an unlimited number of times. If a user is not found in the specified group, a single user element should exist with a status of ‘does not exist’. If there is an error determining the users of a group, a single user element should exist with a status of ‘error’. If a subgroup is not found in the specified group, a single subgroup element should exist with a status of ‘does not exist’. If there is an error determining the subgroups of a group, a single subgroup element should exist with a status of ‘error’.

Extends: oval-sc:ItemType

Child Elements

Table 780: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
group	oval-sc:EntityItemStringType (0..1)	A string the represents the name of a particular group. In Windows, group names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, groups should be identified in the form: “domaingroup name”. For local groups use: “computer namegroup name”. For built-in accounts on the system, use the group name without a domain.
user	oval-sc:EntityItemStringType (0..unbounded)	A string that represents the name of a particular user. In Windows, user names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, users should be identified in the form: “domainuser name”. For local users use: “computer nameuser name”. For built-in accounts on the system, use the user name without a domain.If the specified group has more than one user as a member, then multiple user elements should exist. If the specified group does not contain a single user, then a single user element should exist with a status of ‘does not exist’. If there is an error determining the users that are members of the group, then a single user element should be included with a status of ‘error’.
subgroup	oval-sc:EntityItemStringType (0..unbounded)	A string that represents the name of a particular subgroup in the specified group. In Windows, group names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, the subgroups should be identified in the form: “domaingroup name”. In a local environment, the subgroups should be identified in the form: “computer namegroup name”. If the subgroups are built-in groups, the subgroups should be identified in the form: “group name” without a domain component.If the specified group has more than one subgroup as a member, then multiple subgroup elements should exist. If the specified group does not contain a single subgroup, then a single subgroup element should exist with a status of ‘does not exist’. If there is an error determining the subgroups that are members of the group, then a single subgroup element should be included with a status of ‘error’.

< group_sid_item >

The Windows group_sid_item allows the different users and subgroups, that directly belong to specific groups (identified by SID), to be collected. The collected subgroups will not be resolved to find indirect user or subgroup members. If the subgroups need to be resolved, it should be done using the sid_sid_object. Note that the user and subgroup elements can appear an unlimited number of times. If a user is not found in the specified group, a single user element should exist with a status of ‘does not exist’. If there is an error determining the users of a group, a single user element should exist with a status of ‘error’. If a subgroup is not found in the specified group, a single subgroup element should exist with a status of ‘does not exist’. If there is an error determining the subgroups of a group, a single subgroup element should exist with a status of ‘error’.

Extends: oval-sc:ItemType

Child Elements

Table 781: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
group_sid	oval-sc:EntityItemStringType (0..1)	A string the represents the SID of a particular group.
user_sid	oval-sc:EntityItemStringType (0..unbounded)	A string that represents the SID of a particular user. If the specified group has more than one user as a member, then multiple user_sid entities should exist. If the specified group does not contain a single user, then a single user_sid entity should exist with a status of 'does not exist'. If there is an error determining the userss that are members of the group, then a single user_sid entity should be included with a status of 'error'.
subgroup_sid	oval-sc:EntityItemStringType (0..unbounded)	A string that represents the SID of a particular subgroup. If the specified group has more than one subgroup as a member, then multiple subgroup_sid entities should exist. If the specified group does not contain a single subgroup, a single subgroup_sid entity should exist with a status of 'does not exist'. If there is an error determining the subgroups that are members of the group, then a single subgroup_sid entity should be included with a status of 'error'.

< interface_item >

Enumerate various attributes about the interfaces on a system.

Extends: oval-sc:ItemType

Child Elements

Table 782: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	This element specifies the name of an interface.
index	oval-sc:EntityItemIntType (0..1)	This element specifies index that identifies the interface.
type	win-sc:EntityItemInterfaceTypeType (0..1)	This element specifies the type of interface which is limited to certain set of values.
hardware_addr	oval-sc:EntityItemStringType (0..1)	This element specifies the hardware or MAC address of the physical network card. MAC addresses should be formatted according to the IEEE 802-2001 standard which states that a MAC address is a sequence of six octet values, separated by hyphens, where each octet is represented by two hexadecimal digits. Uppercase letters should also be used to represent the hexadecimal digits A through F.
inet_addr	oval-sc:EntityItemIPAddressStringType (0..1)	This element specifies the IP address of the specific interface. Note that the IP address can be IPv4 or IPv6. If the IP address is an IPv6 address, this entity should be expressed as an IPv6 address prefix using CIDR notation and the netmask entity should not be collected.
broadcast_addr	oval-sc:EntityItemIPAddressStringType (0..1)	This element specifies the broadcast address. A broadcast address is typically the IP address with the host portion set to either all zeros or all ones. Note that the IP address can be IPv4 or IPv6.
netmask	oval-sc:EntityItemIPAddressStringType (0..1)	This element specifies the subnet mask for the IP address. Note that if the inet_addr entity contains an IPv6 address prefix, this entity should not be collected.
addr_type	win-sc:EntityItemAddrTypeType (0..unbounded)	This element specifies the address type or state of a specific interface. Each interface can be associated with more than one value meaning the addr_type element can occur multiple times.

< junction_item >

The junction_item element identifies the result generated for a junction_object.

Extends: oval-sc:ItemType

Child Elements

Table 783: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
path	oval-sc:EntityItemStringType (1..1)	Specifies the path to the subject junction, specified by the junction_object.
canonical_path	oval-sc:EntityItemStringType (1..1)	Specifies the canonical path for the target of the Windows junction specified by the path.
windows_view	win-sc:EntityItemWindowsViewType (0..1)	The windows view value from which this OVAL Item was collected. This is used to indicate from which view (32-bit or 64-bit), the associated Item was collected. A value of '32_bit' indicates the Item was collected from the 32-bit view. A value of '64-bit' indicates the Item was collected from the 64-bit view. Omitting this entity removes any assertion about which view the Item was collected from, and therefore it is strongly suggested that this entity be set.

< license_item >

The license_item element stores the different information that can be found in the Windows license registry value. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-sc:ItemType

Child Elements

Table 784: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	This element describes the name of a license entry.
type	win-sc:EntityItemRegistryTypeType (0..1)	Specifies the type of data stored by the license entry. Valid values are REG_BINARY, REG_DWORD, and REG_SZ. Please refer to the EntityItemRegistryTypeType for more information about the different possible types.
value	oval-sc:EntityItemAnySimpleType (0..1)	The value entity holds the actual value of the specified license entry. The representation of the value as well as the associated datatype attribute depends on type of data stored in the license entry. If the specified license entry is of type REG_BINARY, then the datatype attribute should be set to 'binary' and the data represented by the value entity should follow the xsd:hexBinary form. (each binary octet is encoded as two hex digits) If the registry key is of type REG_DWORD, then the datatype attribute should be set to 'int' and the value entity should represent the data as an integer. If the specified registry key is of type REG_SZ, then the datatype should be 'string' and the value entity should be a copy of the string.

< lockoutpolicy_item >

The lockoutpolicy item enumerates various attributes associated with lockout information for users and global groups in the security database.

Extends: oval-sc:ItemType

Child Elements

Table 785: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
force_logoff	oval-sc:EntityItemIntType (0..1)	Specifies, in seconds (from a DWORD), the amount of time between the end of the valid logon time and the time when the user is forced to log off the network. A value of TIMEQ_FOREVER (max DWORD value, 4294967295) indicates that the user is never forced to log off. A value of zero indicates that the user will be forced to log off immediately when the valid logon time expires. See the USER_MODALS_INFO_0 structure returned by a call to NetUserModalsGet().
lockout_duration	oval-sc:EntityItemIntType (0..1)	Specifies, in seconds, how long a locked account remains locked before it is automatically unlocked. See the USER_MODALS_INFO_3 structure returned by a call to NetUserModalsGet().
lockout_observation_window	oval-sc:EntityItemIntType (0..1)	Specifies the maximum time, in seconds, that can elapse between any two failed logon attempts before lockout occurs. See the USER_MODALS_INFO_3 structure returned by a call to NetUserModalsGet().
lockout_threshold	oval-sc:EntityItemIntType (0..1)	Specifies the number of invalid password authentications that can occur before an account is marked "locked out." See the USER_MODALS_INFO_3 structure returned by a call to NetUserModalsGet().

< metabase_item >

This item gathers information from the specified metabase keys.

Extends: oval-sc:ItemType

Child Elements

Table 786: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
key	oval-sc:EntityItemStringType (0..1)	This element describes a metabase key to be gathered.
id	oval-sc:EntityItemIntType (0..1)	The id element specifies a particular object under the metabase key. If the xsi:nil attribute is set to true, then the item being represented is the higher level metabase key. Using xsi:nil here will result in a status of 'not collected' for the other entities associated with this item since these entities are not associated with a key by itself.
name	oval-sc:EntityItemStringType (0..1)	This element describes the name of the specified metabase object.
user_type	oval-sc:EntityItemStringType (0..1)	The user_type element is an unsigned 32-bit integer (DWORD) that specifies the user type of the data. See the METADATA_RECORD structure.
data_type	oval-sc:EntityItemStringType (0..1)	The data_type element identifies the type of data in the metabase entry. See the METADATA_RECORD structure.
data	oval-sc:EntityItemAnySimpleType (0..unbounded)	The actual data of the named item under the specified metabase key. If the specified metabase key is of type multi string, then multiple value elements should exist to describe the array of strings.

< ntuser_item >

The windows ntuser_item specifies information that can be collected from a particular ntuser.dat file.

Extends: oval-sc:ItemType

Child Elements

Table 787: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
key	oval-sc:EntityItemStringType (0..1)	This element describes a registry key normally found in the HKCU hive to be tested.
name	oval-sc:EntityItemStringType (0..1)	This element describes the name of a registry key. If the xsi:nil attribute is set to true, then the item being represented is the higher level key. Using xsi:nil here will result in a status of 'does not exist' for the type, and value entities since these entities are not associated with a key by itself.
sid	oval-sc:EntityItemStringType (0..1)	This element holds a string that represents the SID of a particular user.
username	oval-sc:EntityItemStringType (0..1)	The username entity holds a string that represents the name of a particular user. In Windows, user names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, users should be identified in the form: "domainuser name". For local users use: "computer nameuser name".
account_type	win-sc:EntityItemNTUserAccountTypeType (0..1)	The account_type element describes if the user account is a local account or domain account.
logged_on	oval-sc:EntityItemBoolType (0..1)	The logged_on element describes if the user account is currently logged on to the computer.
enabled	oval-sc:EntityItemBoolType (0..1)	The enabled element describes if the user account is enabled or disabled.
date_modified	oval-sc:EntityItemIntType (0..1)	Time of last modification of file. The string should represent the FILETIME structure which is a 64-bit value representing the number of 100-nanosecond intervals since January 1, 1601 (UTC).
days_since_modified	oval-sc:EntityItemIntType (0..1)	The number of days since the ntuser.dat file was last modified. The value should be rounded up to the next whole integer.
filepath	oval-sc:EntityItemStringType (0..1)	This element describes the filepath of the ntuser.dat file.
last_write_time	oval-sc:EntityItemIntType (0..1)	The last time that the key or any of its value entries was modified. The value of this entity represents the FILETIME structure which is a 64-bit value representing the number of 100-nanosecond intervals since January 1, 1601 (UTC). Last write time can be queried on a hive, key, or name. When collecting only information about a registry hive the last write time will be the time the hive or any of its entiries was written to. When collecting only information about a registry hive and key the last write time will be the time the key or any of its entiries was written to. When collecting only information about a registry name the last write time will be the time the name was written to. See the RegQueryInfoKey function lpftLastWriteTime.
type	win-sc:EntityItemRegistryTypeType (0..1)	Specifies the type of data stored by the registry key. Please refer to the EntityItemRegistryTypeType for more information about the different possible types.
value	oval-sc:EntityItemAnySimpleType (0..unbounded)	The value entity holds the actual value of the specified registry key. The representation of the value as well as the associated datatype attribute depends on type of data stored in the registry key. If the specified registry key is of type REG_BINARY, then the datatype attribute should be set to 'binary' and the data represented by the value entity should follow the xsd:hexBinary form. (each binary octet is encoded as two hex digits) If the registry key is of type REG_DWORD or REG_QWORD, then the datatype attribute should be set to 'int' and the value entity should represent the data as an integer. If the specified registry key is of type REG_EXPAND_SZ, then the datatype attribute should be set to 'string' and the pre-expanded string should be represented by the value entity. If the specified registry key is of type REG_MULTI_SZ, then multiple value entities should exist to describe the array of strings, with each value element holds a single string. In the end, there should be the same

< passwordpolicy_item >

Specific policy items associated with passwords. It is important to note that these policies are specific to certain versions of Windows. As a result, the documentation for that version of Windows should be consulted for more information. Information is stored in the SAM or Active Directory but is encrypted or hidden so the registry_item and activedirectory_item are of no use. If this can be figured out, then the password_policy item is not needed.

Extends: oval-sc:ItemType

Child Elements

Table 788: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
max_password_age	oval-sc:EntityItemIntType (0..1)	Specifies, in seconds (from a DWORD), the maximum allowable password age. A value of TIMEQ_FOREVER (max DWORD value, 4294967295) indicates that the password never expires. The minimum valid value for this element is ONE_DAY (86400). See the USER_MODALS_INFO_0 structure returned by a call to NetUserModalsGet().
min_password_age	oval-sc:EntityItemIntType (0..1)	Specifies the minimum number of seconds that can elapse between the time a password changes and when it can be changed again. A value of zero indicates that no delay is required between password updates.
min_password_len	oval-sc:EntityItemIntType (0..1)	Specifies the minimum allowable password length. Valid values for this element are zero through PWLEN.
password_hist_len	oval-sc:EntityItemIntType (0..1)	Specifies the length of password history maintained. A new password cannot match any of the previous usrmod0_password_hist_len passwords. Valid values for this element are zero through DEF_MAX_PWHIST.
password_complexity	oval-sc:EntityItemBoolType (0..1)	A boolean value that signifies whether passwords must meet the complexity requirements put forth by the operating system.
reversible_encryption	oval-sc:EntityItemBoolType (0..1)	Determines whether or not passwords are stored using reversible encryption.
anonymous_name_lookup	oval-sc:EntityItemBoolType (0..1)	Determines whether or not an anonymous user may query the local LSA policy.

< peheader_item >

The peheader_item describes the metadata associated with a PE file header. For more information, please see the documentation for the IMAGE_FILE_HEADER and IMAGE_OPTIONAL_HEADER structures.

Extends: oval-sc:ItemType

Child Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-sc:EntityItemStringType (0..1)	The filepath element specifies the absolute
path	oval-sc:EntityItemStringType (0..1)	The path element specifies the directory c
filename	oval-sc:EntityItemStringType (0..1)	The filename element specifies the name
header_signature	oval-sc:EntityItemStringType (0..1)	The header_signature entity is the signatu
target_machine_type	win-sc:EntityItemPeTargetMachineType (0..1)	The target_machine_type entity is an unsi
number_of_sections	oval-sc:EntityItemIntType (0..1)	The number_of_sections entity is an unsig
time_date_stamp	oval-sc:EntityItemIntType (0..1)	The time_date_stamp entity is an unsigne
pointer_to_symbol_table	oval-sc:EntityItemIntType (0..1)	The pointer_to_symbol_table entity is an
number_of_symbols	oval-sc:EntityItemIntType (0..1)	The number_of_symbols entity is an unsig
size_of_optional_header	oval-sc:EntityItemIntType (0..1)	The size_of_optional_header entity is an
image_file_relocs_stripped	oval-sc:EntityItemBoolType (0..1)	The image_file_relocs_stripped entity is a
image_file_executable_image	oval-sc:EntityItemBoolType (0..1)	The image_file_executable_image entity i
image_file_line_nums_stripped	oval-sc:EntityItemBoolType (0..1)	The image_file_line_nums_stripped entity
image_file_local_syms_stripped	oval-sc:EntityItemBoolType (0..1)	The image_file_local_syms_stripped entit
image_file_aggresive_ws_trim	oval-sc:EntityItemBoolType (0..1)	The image_file_aggressive_ws_trim entity
image_file_large_address_aware	oval-sc:EntityItemBoolType (0..1)	The image_file_large_address_aware entit
image_file_16bit_machine	oval-sc:EntityItemBoolType (0..1)	The image_file_16bit_machine entity is a
image_file_bytes_reversed_lo	oval-sc:EntityItemBoolType (0..1)	The image_file_bytes_reversed_lo entity i
image_file_32bit_machine	oval-sc:EntityItemBoolType (0..1)	The image_file_32bit_machine entity is a
image_file_debug_stripped	oval-sc:EntityItemBoolType (0..1)	The image_file_debug_stripped entity is a
image_file_removable_run_from_swap	oval-sc:EntityItemBoolType (0..1)	The image_file_removable_run_from_sw
image_file_system	oval-sc:EntityItemBoolType (0..1)	The image_file_system entity is a boolean
image_file_dll	oval-sc:EntityItemBoolType (0..1)	The image_file_dll entity is a boolean valu
image_file_up_system_only	oval-sc:EntityItemBoolType (0..1)	The image_file_up_system_only entity is
image_file_bytes_revereseed_hi	oval-sc:EntityItemBoolType (0..1)	The image_file_bytes_reversed_hi entity i
magic_number	oval-sc:EntityItemIntType (0..1)	The magic_number entity is an unsigned
major_linker_version	oval-sc:EntityItemIntType (0..1)	The major_linker_version entity is a BYT
minor_linker_version	oval-sc:EntityItemIntType (0..1)	The minor_linker_version entity is a BYT
size_of_code	oval-sc:EntityItemIntType (0..1)	The size_of_code entity is an unsigned 32
size_of_initialized_data	oval-sc:EntityItemIntType (0..1)	The size_of_initialized_data entity is an u
size_of_uninitialized_data	oval-sc:EntityItemIntType (0..1)	The size_of_uninitialized_data entity is a
address_of_entry_point	oval-sc:EntityItemIntType (0..1)	The address_of_entry_point entity is an u
base_of_code	oval-sc:EntityItemIntType (0..1)	The base_of_code entity is an unsigned 3
base_of_data	oval-sc:EntityItemIntType (0..1)	The base_of_data entity is an unsigned 32
image_base_address	oval-sc:EntityItemIntType (0..1)	The image_base_address entity is an unsi
section_alignment	oval-sc:EntityItemIntType (0..1)	The section_alignment entity is an unsign
file_alignment	oval-sc:EntityItemIntType (0..1)	The file_alignment entity is an unsigned 3
major_operating_system_version	oval-sc:EntityItemIntType (0..1)	The major_operating_system_version enti
minor_operating_system_version	oval-sc:EntityItemIntType (0..1)	The minor_operating_system_version ent
major_image_version	oval-sc:EntityItemIntType (0..1)	The major_image_version entity is an uns
minor_image_version	oval-sc:EntityItemIntType (0..1)	The minor_image_version entity is an uns
major_subsystem_version	oval-sc:EntityItemIntType (0..1)	The major_subsystem_version entity is a
minor_susbsystem_version	oval-sc:EntityItemIntType (0..1)	The minor_subsystem_version entity is a
size_of_image	oval-sc:EntityItemIntType (0..1)	The size_of_image entity is an unsigned 3
size_of_headers	oval-sc:EntityItemIntType (0..1)	The size_of_headers entity is an unsigned
checksum	oval-sc:EntityItemIntType (0..1)	The checksum entity is an unsigned 32-bi

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
subsystem	win-sc:EntityItemPeSubsystemType (0..1)	The subsystem entity is an unsigned 32-bi
dll_characteristics	oval-sc:EntityItemIntType (0..unbounded)	The dll_characteristics entity is an unsigne
size_of_stack_reserve	oval-sc:EntityItemIntType (0..1)	The time_date_stamp entity is an unsigne
size_of_stack_commit	oval-sc:EntityItemIntType (0..1)	The time_date_stamp entity is an unsigne
size_of_heap_reserve	oval-sc:EntityItemIntType (0..1)	The time_date_stamp entity is an unsigne
size_of_heap_commit	oval-sc:EntityItemIntType (0..1)	The time_date_stamp entity is an unsigne
loader_flags	oval-sc:EntityItemIntType (0..1)	The loader_flags entity is an unsigned 32-
number_of_rva_and_sizes	oval-sc:EntityItemIntType (0..1)	The number_of_rva_and_sizes entity is ar
real_number_of_directory_entries	oval-sc:EntityItemIntType (0..1)	The real_number_of_directory_entries en
windows_view	win-sc:EntityItemWindowsViewType (0..1)	The windows view value from which this

< port_item >

Information about open listening ports.

Extends: oval-sc:ItemType

Child Elements

Table 790: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
local_address	oval-sc:EntityItemIPAddressStringType (0..1)	This element specifies the local IP address the listening port is bound to. Note that the IP address can be IPv4 or IPv6.
local_port	oval-sc:EntityItemIntType (0..1)	This element specifies the number assigned to the local listening port.
protocol	win-sc:EntityItemProtocolType (0..1)	This element specifies the type of listening port. It is restricted to either TCP or UDP.
pid	oval-sc:EntityItemIntType (0..1)	The id given to the process that is associated with the specified listening port.
foreign_address	oval-sc:EntityItemIPAddressStringType (0..1)	This is the IP address with which the program is communicating, or with which it will communicate, in the case of a listening server. Note that the IP address can be IPv4 or IPv6.
foreign_port	oval-sc:EntityItemStringType (0..1)	This is the TCP or UDP port to which the program communicates.

< printereffectiverights_item >

This item stores the effective rights of a printer that a discretionary access control list (DACL) structure grants to a specified trustee. The trustee's effective rights are determined checking all access-allowed and access-denied access control entries (ACEs) in the DACL. For help with this test see the GetEffectiveRightsFromAcl() api.

Extends: oval-sc:ItemType

Child Elements

Table 791: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
printer_name	oval-sc:EntityItemStringType (0..1)	The printer_name enitity specifies the name of the printer.
trustee_sid	oval-sc:EntityItemStringType (0..1)	The trustee_sid entity specifies the SID that associated a user, group, system, or program (such as a Windows service).
standard_delete	oval-sc:EntityItemBoolType (0..1)	The right to delete the object.
standard_read_control	oval-sc:EntityItemBoolType (0..1)	The right to read the information in the object's security descriptor, not including the information in the SACL.
standard_write_dac	oval-sc:EntityItemBoolType (0..1)	The right to modify the DACL in the object's security descriptor.
standard_write_owner	oval-sc:EntityItemBoolType (0..1)	The right to change the owner in the object's security descriptor.
standard_synchronize	oval-sc:EntityItemBoolType (0..1)	The right to use the object for synchronization. This enables a thread to wait until the object is in the signaled state. Some object types do not support this access right.
access_system_security	oval-sc:EntityItemBoolType (0..1)	Indicates access to a system access control list (SACL).
generic_read	oval-sc:EntityItemBoolType (0..1)	Read access.
generic_write	oval-sc:EntityItemBoolType (0..1)	Write access.
generic_execute	oval-sc:EntityItemBoolType (0..1)	Execute access.
generic_all	oval-sc:EntityItemBoolType (0..1)	Read, write, and execute access.
printer_access_administer	oval-sc:EntityItemBoolType (0..1)	
printer_access_use	oval-sc:EntityItemBoolType (0..1)	
job_access_administer	oval-sc:EntityItemBoolType (0..1)	
job_access_read	oval-sc:EntityItemBoolType (0..1)	

< process_item >

Information about running processes.

Extends: oval-sc:ItemType

Child Elements

Table 792: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
command_line	oval-sc:EntityItemStringType (0..1)	The command_line entity is the string used to start the process. This includes any parameters that are part of the command line.
pid	oval-sc:EntityItemIntType (0..1)	The id given to the process that is created for a specified command line.
ppid	oval-sc:EntityItemIntType (0..1)	The id given to the parent of the process that is created for the specified command line
priority	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	The base priority of the process. The priority value range is from 0 to 31.
image_path	oval-sc:EntityItemStringType (0..1)	The image_path entity represents the name of the executable file for the process.
current_dir	oval-sc:EntityItemStringType (0..1)	The current_dir entity represents the current path to the executable file for the process.
creation_time	oval-sc:EntityItemIntType (0..1)	The creation_time entity represents the creation time of the process. The value of this entity represents the FILETIME structure which is a 64-bit value representing the number of 100-nanosecond intervals since January 1, 1601 (UTC). See the GetProcessTimes function lpCreationTime.
dep_enabled	oval-sc:EntityItemBoolType (0..1)	The dep_enabled entity represents whether or not data execution prevention (DEP) is enabled. See the GetProcessDEPPolicy function lpFlags.
primary_window_text	oval-sc:EntityItemStringType (0..1)	The primary_window_text entity represents the title of the primary window of the process. See the GetWindowText function.
name	oval-sc:EntityItemStringType (0..1)	The name of the process.

< registry_item >

The windows registry item specifies information that can be collected about a particular registry key.

Extends: oval-sc:ItemType

Child Elements

Table 793: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
hive	win-sc:EntityItemRegistryHiveType (0..1)	The hive that the registry key belongs to.
key	oval-sc:EntityItemStringType (0..1)	This element describes a registry key to be gathered. Note that the hive portion of the string should not be included, as this data can be found under the hive element. If the xsi:nil attribute is set to true, then the item being represented is the higher level hive or lower level name. Using xsi:nil here will result in a status of 'not collected' for this entity since the item is specific to a hive or name.
name	oval-sc:EntityItemStringType (0..1)	This element describes the name of a registry key. If the xsi:nil attribute is set to true, then the item being represented is the higher level key or hive. Using xsi:nil here will result in a status of 'not collected' since the item is specific to a key or hive.
last_write_time	oval-sc:EntityItemIntType (0..1)	The last time that the key or any of its value entries were modified. The value of this entity represents the FILETIME structure which is a 64-bit value representing the number of 100-nanosecond intervals since January 1, 1601 (UTC). Last write time can be queried on any key, with hives being classified as a type of key. When collecting only information about a registry hive or key the last write time will be the time the key or any of its entries were modified. When collecting only information about a registry name the last write time will be the time the containing key was modified. Thus when collecting information about a registry name, the last write time does not correlate directly to the specified name. See the RegQueryInfoKey function lpftLastWriteTime.
type	win-sc:EntityItemRegistryTypeType (0..1)	Specifies the type of data stored by the registry key. Please refer to the EntityItemRegistryTypeType for more information about the different possible types.
value	oval-sc:EntityItemAnySimpleType (0..unbounded)	The value entity holds the actual value of the specified registry key. The representation of the value as well as the associated datatype attribute depends on type of data stored in the registry key. If the value being tested is of type REG_BINARY, then the datatype attribute should be set to 'binary' and the data represented by the value entity should follow the xsd:hexBinary form. (each binary octet is encoded as two hex digits) If the value being tested is of type REG_DWORD, REG_QWORD, REG_DWORD_LITTLE_ENDIAN, REG_DWORD_BIG_ENDIAN, or REG_QWORD_LITTLE_ENDIAN then the datatype attribute should be set to 'int' and the value entity should represent the data as an unsigned integer. DWORD and QWORD values represnt unsigned 32-bit and 64-bit integers, respectively. If the value being tested is of type REG_EXPAND_SZ, then the datatype attribute should be set to 'string' and the pre-expanded string should be represented by the value entity. If the value being tested is of type REG_MULTI_SZ, then only a single string (one of the multiple strings) should be tested using the value entity with the datatype attribute set to 'string'. In order to test multiple values, multiple OVAL registry tests or multiple states should be combined. Reg_multi_sz values, with no values, should be given a status of "does not exist". If the specified registry key is of type REG_SZ, then the datatype should be 'string' and the value entity should be a copy of the string. If the value being tested is of type REG_LINK, then the datatype attribute should be set to 'string' and the null-terminated Unicode string should be represented by the value entity.
expanded_value	oval-sc:EntityItemAnySimpleType (0..1)	For registry values of type REG_EXPAND_SZ, this entity contains the expanded value. Otherwise, this entity should not exist.
windows_view	win-sc:EntityItemWindowsViewType (0..1)	The windows view value from which this OVAL Item was collected. This is used to indicate from which view (32-bit or 64-bit), the associated Item was collected. A value of '32_bit' indicates the Item was collected from the 32-bit view. A value of '64-bit' indicates the Item was collected from the 64-bit view. Omitting this entity removes any assertion about which view the Item was collected from, and therefore it is strongly suggested that this entity be set.

< regkeyauditedpermissions_item >

This item stores the audited access rights of a registry key that a system access control list (SACL) structure grants to a specified trustee. The trustee's audited access rights are determined checking all access control entries (ACEs) in the SACL. For help with this test see the GetAuditedPermissionsFromAcl() api.

Extends: oval-sc:ItemType

Child Elements

Table 794: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
hive	win-sc:EntityItemRegistryHiveType (0..1)	This element specifies the hive of a registry key on the machine from which the SACL was retrieved.
key	oval-sc:EntityItemStringType (0..1)	This element specifies a registry key on the machine from which the SACL was retrieved. Note that the hive portion of the string should not be inclueded, as this data should be found under the hive element.
trustee_sid	oval-sc:EntityItemStringType (0..1)	The security identifier (SID) of the specified trustee name.
trustee_name (Deprecated)	oval-sc:EntityItemStringType (0..1)	This element specifies the trustee name associated with this particular DACL. A trustee can be a user, group, or program (such as a Windows service). In Windows, trustee names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, trustee names should be identified in the form: “domaintrustee name”. For local trustee names use: “computer nametrustee name”. For built-in accounts on the system, use the trustee name without a domain.
standard_delete	win-sc:EntityItemAuditType (0..1)	The right to delete the object.
standard_read_control	win-sc:EntityItemAuditType (0..1)	The right to read the information in the object’s security descriptor, not including the information in the SACL.
standard_write_dac	win-sc:EntityItemAuditType (0..1)	The right to modify the DACL in the object’s security descriptor.
standard_write_owner	win-sc:EntityItemAuditType (0..1)	The right to change the owner in the object’s security descriptor.
standard_synchronize (Deprecated)	win-sc:EntityItemAuditType (0..1)	The right to use the object for synchronization. This enables a thread to wait until the object is in the signaled state. Some object types do not support this access right.
access_system_security	win-sc:EntityItemAuditType (0..1)	Indicates access to a system access control list (SACL).
generic_read	win-sc:EntityItemAuditType (0..1)	Read access.
generic_write	win-sc:EntityItemAuditType (0..1)	Write access.
generic_execute	win-sc:EntityItemAuditType (0..1)	Execute access.
generic_all	win-sc:EntityItemAuditType (0..1)	Read, write, and execute access.
key_query_value	win-sc:EntityItemAuditType (0..1)	
key_set_value	win-sc:EntityItemAuditType (0..1)	

< regkeyeffectiverights_item >

This item stores the effective rights of a registry key that a discretionary access control list (DACL) structure grants to a specified trustee. The trustee's effective rights are determined checking all access-allowed and access-denied access control entries (ACEs) in the DACL. For help with this test see the GetEffectiveRightsFromAcl() api.

Extends: oval-sc:ItemType

Child Elements

Table 795: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
hive	win-sc:EntityItemRegistryHiveType (0..1)	The hive that the registry key belongs to.
key	oval-sc:EntityItemStringType (0..1)	This element describes a registry key to be gathered. Note that the hive portion of the string should not be inclueded, as this data can be found under the hive element. If the xsi:nil attribute is set to true, then the item being represented is the higher level hive.
trustee_sid	oval-sc:EntityItemStringType (0..1)	The trustee_sid entity specifies the SID that associated a user, group, system, or program (such as a Windows service).
trustee_name (Deprecated)	oval-sc:EntityItemStringType (0..1)	This element specifies the trustee name associated with this particular DACL. A trustee can be a user, group, or program (such as a Windows service). In Windows, trustee names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, trustee names should be identified in the form: "domaintrustee name". For local trustee names use: "computer nametrustee name". For built-in accounts on the system, use the trustee name without a domain.
standard_delete	oval-sc:EntityItemBoolType (0..1)	The right to delete the object.
standard_read_control	oval-sc:EntityItemBoolType (0..1)	The right to read the information in the object's security descriptor, not including the information in the SACL.
standard_write_dac	oval-sc:EntityItemBoolType (0..1)	The right to modify the DACL in the object's security descriptor.
standard_write_owner	oval-sc:EntityItemBoolType (0..1)	The right to change the owner in the object's security descriptor.
standard_synchronize (Deprecated)	oval-sc:EntityItemBoolType (0..1)	The right to use the object for synchronization. This enables a thread to wait until the object is in the signaled state. Some object types do not support this access right.
access_system_security	oval-sc:EntityItemBoolType (0..1)	Indicates access to a system access control list (SACL).
generic_read	oval-sc:EntityItemBoolType (0..1)	Read access.
generic_write	oval-sc:EntityItemBoolType (0..1)	Write access.
generic_execute	oval-sc:EntityItemBoolType (0..1)	Execute access.
generic_all	oval-sc:EntityItemBoolType (0..1)	Read, write, and execute access.
key_query_value	oval-sc:EntityItemBoolType (0..1)	
key_set_value	oval-sc:EntityItemBoolType (0..1)	

< service_item >

This item stores information about Windows services that are present on the system.

Extends: oval-sc:ItemType

Child Elements

Table 796: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
service_name	oval-sc:EntityItemStringType (0..1)	The service_name element specifies the name of the service as specified in the Service Control Manager (SCM) database.
display_name	oval-sc:EntityItemStringType (0..1)	The display_name element specifies the name of the service as specified in tools such as Control Panel->Administrative Tools->Services.
description	oval-sc:EntityItemStringType (0..1)	The description element specifies the description of the service.
service_type	win-sc:EntityItemServiceTypeType (0..unbounded)	The service_type element specifies the type of the service.
start_type	win-sc:EntityItemServiceStartTypeType (0..1)	The start_type element specifies when the service should be started.
current_state	win-sc:EntityItemServiceCurrentStateType (0..1)	The current_state element specifies the current state of the service.
controls_accepted	win-sc:EntityItemServiceControlsAcceptedType (0..unbounded)	The controls_accepted element specifies the control codes that a service will accept and process.
start_name	oval-sc:EntityItemStringType (0..1)	The start_name element specifies the account under which the process should run.
path	oval-sc:EntityItemStringType (0..1)	The path element specifies the path to the binary of the service.
pid	oval-sc:EntityItemIntType (0..1)	The pid element specifies the process ID of the service.
service_flag	oval-sc:EntityItemBoolType (0..1)	The service_flag element specifies if the service is in a system process that must always run (1) or if the service is in a non-system process or is not running (0). If the service is not running, the pid will be 0. Otherwise, the pid will be non-zero.
dependencies	oval-sc:EntityItemStringType (0..unbounded)	The dependencies element specifies the dependencies of this service on other services.

< serviceeffectiverights_item >

This item stores the effective rights of a service that a discretionary access control list (DACL) structure grants to a specified trustee. The trustee's effective rights are determined by checking all access-allowed and access-denied access control entries (ACEs) in the DACL. For help with this test see the GetEffectiveRightsFromAcl() api.

Extends: oval-sc:ItemType

Child Elements

Table 797: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
service_name	oval-sc:EntityItemStringType (0..1)	The service_name element specifies a service on the machine from which to retrieve the DACL. Note that the service_name element should contain the actual name of the service and not its display name that is found in Control Panel->Administrative Tools->Services. For example, if you wanted to check the effective rights of the Automatic Updates service you would specify ‘wuauserv’ for the service_name element not ‘Automatic Updates’.
trustee_sid	oval-sc:EntityItemStringType (0..1)	The trustee_sid element specifies the SID that is associated with a user, group, system, or program (such as a Windows service).
standard_delete	oval-sc:EntityItemBoolType (0..1)	This permission is required to call the DeleteService function to delete the service.
standard_read_control	oval-sc:EntityItemBoolType (0..1)	This permission is required to call the QueryServiceObjectSecurity function to query the security descriptor of the service object.
standard_write_dac	oval-sc:EntityItemBoolType (0..1)	This permission is required to call the SetServiceObjectSecurity function to modify the Dacl member of the service object’s security descriptor.
standard_write_owner	oval-sc:EntityItemBoolType (0..1)	This permission is required to call the SetServiceObjectSecurity function to modify the Owner and Group members of the service object’s security descriptor.
generic_read	oval-sc:EntityItemBoolType (0..1)	Read access (STANDARD_RIGHTS_READ, SERVICE_QUERY_CONFIG, SERVICE_QUERY_STATUS, SERVICE_INTERROGATE, SERVICE_ENUMERATE_DEPENDENTS).
generic_write	oval-sc:EntityItemBoolType (0..1)	Write access (STANDARD_RIGHTS_WRITE, SERVICE_CHANGE_CONFIG).
generic_execute	oval-sc:EntityItemBoolType (0..1)	Execute access (STANDARD_RIGHTS_EXECUTE, SERVICE_START, SERVICE_STOP, SERVICE_PAUSE_CONTINUE, SERVICE_USER_DEFINED_CONTROL).
service_query_conf	oval-sc:EntityItemBoolType (0..1)	This permission is required to call the QueryServiceConfig and QueryServiceConfig2 functions to query the service configuration.
service_change_conf	oval-sc:EntityItemBoolType (0..1)	This permission is required to call the ChangeServiceConfig or ChangeServiceConfig2 function to change the service configuration.
service_query_stat	oval-sc:EntityItemBoolType (0..1)	This permission is required to call the QueryServiceStatusEx function to ask the service control manager about the status of the service.
service_enum_dependents	oval-sc:EntityItemBoolType (0..1)	This permission is required to call the EnumDependentServices function to enumerate all the services dependent on the service.
service_start	oval-sc:EntityItemBoolType (0..1)	This permission is required to call the StartService function to start the service.
service_stop	oval-sc:EntityItemBoolType (0..1)	This permission is required to call the ControlService function to stop the service.
service_pause	oval-sc:EntityItemBoolType (0..1)	This permission is required to call the ControlService function to pause or continue the service.
service_interrogate	oval-sc:EntityItemBoolType (0..1)	This permission is required to call the ControlService function to ask the service to report its status immediately.

< sharedresource_item >

Extends: oval-sc:ItemType

Child Elements

Table 798: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
netname	oval-sc:EntityItemStringType (0..1)	The share name of the resource.
shared_type	win-sc:EntityItemSharedResourceTypeType (0..1)	The type of the shared resource.
max_uses	oval-sc:EntityItemIntType (0..1)	The maximum number of concurrent connections that the shared resource can accommodate.
current_uses	oval-sc:EntityItemIntType (0..1)	The number of current connections to the shared resource.
local_path	oval-sc:EntityItemStringType (0..1)	The local path for the shared resource.
access_read_permission	oval-sc:EntityItemBoolType (0..1)	Permission to read data from a resource and, by default, to execute the resource.
access_write_permission	oval-sc:EntityItemBoolType (0..1)	Permission to write data to the resource.
access_create_permission	oval-sc:EntityItemBoolType (0..1)	Permission to create an instance of the resource (such as a file); data can be written to the resource as the resource is created.
access_exec_permission	oval-sc:EntityItemBoolType (0..1)	Permission to execute the resource.
access_delete_permission	oval-sc:EntityItemBoolType (0..1)	Permission to delete the resource.
access_atrib_permission	oval-sc:EntityItemBoolType (0..1)	Permission to modify the resource's attributes (such as the date and time when a file was last modified).
access_perm_permission	oval-sc:EntityItemBoolType (0..1)	Permission to modify the permissions (read, write, create, execute, and delete) assigned to a resource for a user or application.
access_all_permission	oval-sc:EntityItemBoolType (0..1)	Permission to read, write, create, execute, and delete resources, and to modify their attributes and permissions.

< sharedresourceauditedpermissions_item >

This item stores the audited access rights of a shared resource that a system access control list (SACL) structure grants to a specified trustee. The trustee's audited access rights are determined checking all access control entries (ACEs) in the SACL.

Extends: oval-sc:ItemType

Child Elements

Table 799: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
netname	oval-sc:EntityItemStringType (0..1)	The netname entity specifies the name associated with a particular shared resource.
trustee_sid	oval-sc:EntityItemStringType (0..1)	The trustee_sid entity specifies the SID that associated a user, group, system, or program (such as a Windows service).
standard_delete	win-sc:EntityItemAuditType (0..1)	The right to delete the object.
standard_read_control	win-sc:EntityItemAuditType (0..1)	The right to read the information in the object's security descriptor, not including the information in the SACL.
standard_write_dac	win-sc:EntityItemAuditType (0..1)	The right to modify the DACL in the object's security descriptor.
standard_write_owner	win-sc:EntityItemAuditType (0..1)	The right to change the owner in the object's security descriptor.
standard_synchronize	win-sc:EntityItemAuditType (0..1)	The right to use the object for synchronization. This enables a thread to wait until the object is in the signaled state. Some object types do not support this access right.
access_system_security	win-sc:EntityItemAuditType (0..1)	Indicates access to a system access control list (SACL).
generic_read	win-sc:EntityItemAuditType (0..1)	Read access.
generic_write	win-sc:EntityItemAuditType (0..1)	Write access.
generic_execute	win-sc:EntityItemAuditType (0..1)	Execute access.
generic_all	win-sc:EntityItemAuditType (0..1)	Read, write, and execute access.

< sharedresourceeffectiverights_item >

This item stores the effective rights of a shared resource that a discretionary access control list (DACL) structure grants to a specified trustee. The trustee's effective rights are determined checking all access-allowed and access-denied access control entries (ACEs) in the DACL.

Extends: oval-sc:ItemType

Child Elements

Table 800: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
netname	oval-sc:EntityItemStringType (0..1)	The netname entity specifies the name associated with a particular shared resource.
trustee_sid	oval-sc:EntityItemStringType (0..1)	The trustee_sid entity specifies the SID that associated a user, group, system, or program (such as a Windows service).
standard_delete	oval-sc:EntityItemBoolType (0..1)	The right to delete the object.
standard_read_control	oval-sc:EntityItemBoolType (0..1)	The right to read the information in the object's security descriptor, not including the information in the SACL.
standard_write_dac	oval-sc:EntityItemBoolType (0..1)	The right to modify the DACL in the object's security descriptor.
standard_write_owner	oval-sc:EntityItemBoolType (0..1)	The right to change the owner in the object's security descriptor.
standard_synchronize	oval-sc:EntityItemBoolType (0..1)	The right to use the object for synchronization. This enables a thread to wait until the object is in the signaled state. Some object types do not support this access right.
access_system_security	oval-sc:EntityItemBoolType (0..1)	Indicates access to a system access control list (SACL).
generic_read	oval-sc:EntityItemBoolType (0..1)	Read access.
generic_write	oval-sc:EntityItemBoolType (0..1)	Write access.
generic_execute	oval-sc:EntityItemBoolType (0..1)	Execute access.
generic_all	oval-sc:EntityItemBoolType (0..1)	Read, write, and execute access.

< sid_item >

Extends: oval-sc:ItemType

Child Elements

Table 801: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
trustee_name	oval-sc:EntityItemStringType (0..1)	This element specifies the trustee name associated with a particular SID. In Windows, trustee names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, trustee names should be identified in the form: “domaintrustee name”. For local trustee names use: “computer nametrustee name”. For built-in accounts on the system, use the trustee name without a domain.
trustee_sid	oval-sc:EntityItemStringType (0..1)	The security identifier (SID) of the specified trustee name.
trustee_domain	oval-sc:EntityItemStringType (0..1)	The domain of the specified trustee name.

< sid_sid_item >

Extends: oval-sc:ItemType

Child Elements

Table 802: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
trustee_sid	oval-sc:EntityItemStringType (0..1)	The security identifier (SID) of the specified trustee name.
trustee_name	oval-sc:EntityItemStringType (0..1)	This element specifies the trustee name associated with a particular SID. In Windows, trustee names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, trustee names should be identified in the form: “domaintrustee name”. For local trustee names use: “computer nametrustee name”. For built-in accounts on the system, use the trustee name without a domain.
trustee_domain	oval-sc:EntityItemStringType (0..1)	The domain of the specified trustee name.

< systemmetric_item >

The system metric item stores the value of a particular Windows system metric.

Extends: oval-sc:ItemType

Child Elements

Table 803: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
index	win-sc:EntityItemSystemMetricIndexType (0..1)	This element describes the index of a system metric entry.
value	oval-sc:EntityItemIntType (0..1)	The value entity holds the actual value of the specified system metric index.

< uac_item >

The uac_item is used to hold information about settings related to User Access Control within Windows.

Extends: oval-sc:ItemType

Child Elements

Table 804: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
admin_approval_mode	oval-sc:EntityItemBoolType (0..1)	Admin Approval Mode for the Built-in Administrator account.
elevation_prompt_admin	oval-sc:EntityItemStringType (0..1)	Behavior of the elevation prompt for administrators in Admin Approval Mode.
elevation_prompt_standard	oval-sc:EntityItemStringType (0..1)	Behavior of the elevation prompt for standard users.
detect_installations	oval-sc:EntityItemBoolType (0..1)	Detect application installations and prompt for elevation.
elevate_signed_executables	oval-sc:EntityItemBoolType (0..1)	Only elevate executables that are signed and validated.
elevate_uiaccess	oval-sc:EntityItemBoolType (0..1)	Only elevate UIAccess applications that are installed in secure locations.
run_admins_aam	oval-sc:EntityItemBoolType (0..1)	Run all administrators in Admin Approval Mode.
secure_desktop	oval-sc:EntityItemBoolType (0..1)	Switch to the secure desktop when prompting for elevation.
virtualize_write_failures	oval-sc:EntityItemBoolType (0..1)	Virtualize file and registry write failures to per-user locations.

< user_item > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.11
- Reason: Replaced by the user_sid_item. This item uses trustee names for identifying accounts on the system. Trustee names are not unique and the user_sid_item, which uses trustee SIDs which are unique, should be used instead. See the user_sid_item.
- Comment: This object has been deprecated and may be removed in a future version of the language.

The windows user_item allows the different groups (identified by name) that a user belongs to be collected.

Extends: oval-sc:ItemType

Child Elements

Table 805: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
user	oval-sc:EntityItemStringType (0..1)	A string the represents the name of a particular user. In Windows, user names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, users should be identified in the form: "domainuser name". For local users use: "computer_nameuser_name". For built-in accounts on the system, use the user name without a domain.
enabled	oval-sc:EntityItemBoolType (0..1)	A boolean that represents whether the particular user is enabled or not.
group	oval-sc:EntityItemStringType (0..unbounded)	A string that represents the name of a particular group. In Windows, group names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, groups should be identified in the form: "domaingroup name". For local groups use: "computer namegroup name". For built-in accounts on the system, use the group name without a domain.If the specified user belongs to more than one group, then multiple group elements should exist. If the specified user is not a member of a single group, then a single group element should exist with a status of 'does not exist'. If there is an error determining the groups that the user belongs to, then a single group element should be included with a status of 'error'.
last_logon	oval-sc:EntityItemIntType (0..1)	The date and time when the last logon occurred. This value is stored as the number of seconds that have elapsed since 00:00:00, January 1, 1970, GMT. If the target system is a domain controller, this data is maintained separately on each backup domain controller (BDC) in the domain. To obtain an accurate value, you must query each BDC in the domain. The last logoff occurred at the time indicated by the largest retrieved value.
full_name	oval-sc:EntityItemStringType (0..1)	A Unicode string that contains the full name of the user. This string can be a NULL string, or it can have any number of characters before the terminating null character.
comment	oval-sc:EntityItemStringType (0..1)	A Unicode string that contains a comment to associate with the user account. The string can be a NULL string, or it can have any number of characters before the terminating null character.
password_age_days	oval-sc:EntityItemIntType (0..1)	The number of full days that have elapsed since the password was last changed, meaning data partially accumulated should be truncated. Ex: 89.5 days = 89, 90.01 = 90
lockout	oval-sc:EntityItemBoolType (0..1)	The account is currently locked out.
passwd_notreqd	oval-sc:EntityItemBoolType (0..1)	No password is required.
dont_expire_passwd	oval-sc:EntityItemBoolType (0..1)	The password should never expire on the account.
encrypted_text_password_allowed	oval-sc:EntityItemBoolType (0..1)	The user's password is stored under reversible encryption in the Active Directory.
not_delegated	oval-sc:EntityItemBoolType (0..1)	Marks the account as "sensitive"; other users cannot act as delegates of this user account.
use_des_key_only	oval-sc:EntityItemBoolType (0..1)	Restrict this principal to use only Data Encryption Standard (DES) encryption types for keys.
dont_require_preauth	oval-sc:EntityItemBoolType (0..1)	This account does not require Kerberos preauthentication for logon.
pass-	oval-	The password expiration information. Zero if the password has not expired (and nonzero if it has).

< user_sid_item >

The windows user_sid_item allows the different groups (identified by SID) that a user belongs to be collected.

Extends: oval-sc:ItemType

Child Elements

Table 806: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
user_sid	oval-sc:EntityItemStringType (0..1)	A string the represents the SID of a particular user.
enabled	oval-sc:EntityItemBoolType (0..1)	A boolean that represents whether the particular user is enabled or not.
group_sid	oval-sc:EntityItemStringType (0..unbounded)	A string that represents the SID of a particular group. If the specified user belongs to more than one group, then multiple group_sid elements should exist. If the specified user is not a member of a single group, then a single group_sid element should exist with a status of 'does not exist'. If there is an error determining the groups that the user belongs to, then a single group_sid element should be included with a status of 'error'.
last_logon	oval-sc:EntityItemIntType (0..1)	The date and time when the last logon occurred. This value is stored as the number of seconds that have elapsed since 00:00:00, January 1, 1970, GMT.

< userright_item >

Extends: oval-sc:ItemType

Child Elements

Table 807: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
userright	win-sc:EntityItemUserRightType (0..1)	The userright entity holds a string that represents the name of a particular user right/privilege.
trustee_name	oval-sc:EntityItemStringType (0..1)	The trustee_name entity is the unique name associated with the SID that has been granted the specified user right/privilege. A trustee can be a user, group, or program (such as a Windows service). In Windows, trustee names are case-insensitive. As a result, it is recommended that the case-insensitive operations are used for this entity. In a domain environment, trustee names should be identified in the form: “domaintrustee name”. For local trustee names use: “computer nametrustee name”. For built-in accounts on the system, use the trustee name without a domain.
trustee_sid	oval-sc:EntityItemStringType (0..1)	The trustee_sid entity identifies the SID that has been granted the specified user right/privilege.

< volume_item >

The volume item enumerates various attributes about a particular volume mounted to a machine. This includes the various system flags returned by GetVolumeInformation(). It is important to note that these system flags are specific to certain versions of Windows. As a result, the documentation for that version of Windows should be consulted for more information.

Extends: oval-sc:ItemType

Child Elements

Table 808: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
root-path	oval-sc:EntityItemStringType (0..1)	A string that contains the root directory of the volume to be described. A trailing backslash is required. For example, you would specify \MyServerMyShare as "\MyServerMyShare", or the C drive as "C:".
file_system	oval-sc:EntityItemStringType (0..1)	The type of filesystem. For example FAT or NTFS.
name	oval-sc:EntityItemStringType (0..1)	The name of the volume.
drive_type	win-sc:EntityItemDriveTypeType (0..1)	The drive type of the volume.
volume_max_component_length	oval-sc:EntityItemIntType (0..1)	The volume_max_component_length element specifies the maximum length, in TCHARs, of a file name component that a specified file system supports. A file name component is the portion of a file name between backslashes. The value that is stored in the variable that *lpMaximumComponentLength points to is used to indicate that a specified file system supports long names. For example, for a FAT file system that supports long names, the function stores the value 255, rather than the previous 8.3 indicator. Long names can also be supported on systems that use the NTFS file system.
serial_number	oval-sc:EntityItemIntType (0..1)	The volume serial number.
file_case_sensitive_search	oval-sc:EntityItemBoolType (0..1)	The file system supports case-sensitive file names.
file_case_preserved_names	oval-sc:EntityItemBoolType (0..1)	The file system preserves the case of file names when it places a name on disk.
file_unicode_on_disk	oval-sc:EntityItemBoolType (0..1)	The file system supports Unicode in file names as they appear on disk.
file_persistent_acls	oval-sc:EntityItemBoolType (0..1)	The file system preserves and enforces ACLs. For example, NTFS preserves and enforces ACLs, and FAT does not.
file_file_compression	oval-sc:EntityItemBoolType (0..1)	The file system supports file-based compression.
file_volume_quotas	oval-sc:EntityItemBoolType (0..1)	The file system supports disk quotas.
file_supports_sparse_files	oval-sc:EntityItemBoolType (0..1)	The file system supports sparse files.
file_supports_reparse_points	oval-sc:EntityItemBoolType (0..1)	The file system supports reparse points.
file_supports_remote_storage	oval-sc:EntityItemBoolType (0..1)	The file system supports remote storage.
file_volume_is_compressed	oval-sc:EntityItemBoolType (0..1)	The specified volume is a compressed volume; for example, a DoubleSpace volume.

< wmi_item > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.7
- Reason: Replaced by the wmi57_item. This item allows for single fields to be selected from WMI. A new item was created to allow more than one field to be selected in one statement. See the wmi57_item.
- Comment: This object has been deprecated and may be removed in a future version of the language.

The wmi_item outlines information to be checked through Microsoft's WMI interface.

Extends: oval-sc:ItemType

Child Elements

Table 809: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
namespace	oval-sc:EntityItemStringType (0..1)	The WMI namespaces of the specific object.
wql	oval-sc:EntityItemStringType (0..1)	A WQL query used to identify the object(s) specified. Any valid WQL query is allowed with one exception, at most one field is allowed in the SELECT portion of the query. For example SELECT name FROM ... is valid, as is SELECT 'true' FROM ..., but SELECT name, number FROM ... is not valid. This is because the result element in the data section is only designed to work against a single field.
result	oval-sc:EntityItemAnySimpleType (0..unbounded)	The result element specifies how to test objects in the result set of the specified WQL statement. Only one comparable field is allowed. So if the WQL statement look like 'SELECT name FROM ...', then a result element with a value of 'Fred' would test that value against the names returned by the WQL statement. If the WQL statement returns more than one instance of the specified field, then multiple result elements should exist to describe each instance.

< wmi57_item >

The wmi57_item outlines information to be checked through Microsoft's WMI interface.

Extends: oval-sc:ItemType

Child Elements

Table 810: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
namespace	oval-sc:EntityItemStringType (0..1)	The WMI namespaces of the specific object.
wql	oval-sc:EntityItemStringType (0..1)	A WQL query used to identify the object(s) specified. Any valid WQL query is allowed with one exception, all fields must be named. For example SELECT name, age FROM … is valid, but SELECT * FROM … is not valid. This is because the record entity supports only named fields.
result	oval-sc:EntityItemRecordType (0..unbounded)	The result entity holds the results of the specified WQL statement.

< wuaupdatesearcher_item >

The wuaupdatesearcher_item outlines information defined through the Search method of the IUpdateSearcher interface as part of Microsoft's WUA (Windows Update Agent) API. This information is related to the current patch level in a Windows environment. The test extends the standard ItemType as defined in the oval-system-characteristics schema and one should refer to the ItemType description for more information.

Extends: oval-sc:ItemType

Child Elements

Table 811: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
search_criteria	oval-sc:EntityItemStringType (0..1)	
update_id	oval-sc:EntityItemStringType (0..unbounded)	The update_id entity specifies a string that represents a revision-independent identifier of an update. This information is part of the IUpdateIdentity interface that is part of the result of the IUpdateSearcher interface's Search method. Note that multiple update identifiers can be associated with a give search criteria and thus multiple entities can exist for this item.

== EntityItemAddrTypeType ==

The EntityItemAddrTypeType restricts a string value to a specific set of values that describe the different address types of interfaces. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 812: Enumeration Values

Value	Description
MIB_IPADDR_DELETED	The stated IP address is being deleted. The unsigned short value that this corresponds to is 0x0040
MIB_IPADDR_DISCONNECTED	The stated IP address is on a disconnected interface. The unsigned short value that this corresponds to is 0x0008.
MIB_IPADDR_DYNAMIC	The stated IP address is a dynamic IP address. The unsigned short value that this corresponds to is 0x0004.
MIB_IPADDR_PRIMARY	The stated IP address is a primary IP address. The unsigned short value that this corresponds to is 0x0001.
MIB_IPADDR_TRANSIENT	The stated IP address is a transient IP address. The unsigned short value that this corresponds to is 0x0080
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemAdstypeType ==

The EntityItemAdstypeType restricts a string value to a specific set of values that describe the possible types associated with an Active Directory attribute. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 813: Enumeration Values

Value	Description
ADSTYPE_INVALID	The data type is invalid.
ADSTYPE_DN_STRING	The string is of Distinguished Name (path) of a directory service object.
ADSTYPE_CASE_EXACT_STRING	The string is of the case-sensitive type.
ADSTYPE_CASE_IGNORE_STRING	The string is of the case-insensitive type.
ADSTYPE_PRINTABLE_STRING	The string is displayable on the screen or in print.
ADSTYPE_NUMERIC_STRING	The string is of a numeric value to be interpreted as text.
ADSTYPE_BOOLEAN	The data is of a Boolean value.
ADSTYPE_INTEGER	The data is of an integer value.
ADSTYPE_OCTET_STRING	The string is of a byte array.
ADSTYPE_UTC_TIME	The data is of the universal time as expressed in Universal Time Coordinate (UTC).
ADSTYPE_LARGE_INTEGER	The data is of a long integer value.
ADSTYPE_PROV_SPECIFIC	The string is of a provider-specific string.
ADSTYPE_OBJECT_CLASS	Not used.
ADSTYPE_CASEIGNORE_LIST	The data is of a list of case insensitive strings.
ADSTYPE_OCTET_LIST	The data is of a list of octet strings.

Continued on next page

Table 813 – continued from previous page

Value	Description
ADSTYPE_PATH	The string is of a directory path.
ADSTYPE_POSTALADDRESS	The string is of the postal address type.
ADSTYPE_TIMESTAMP	The data is of a time stamp in seconds.
ADSTYPE_BACKLINK	The string is of a back link.
ADSTYPE_TYPEDNAME	The string is of a typed name.
ADSTYPE_HOLD	The data is of the Hold data structure.
ADSTYPE_NETADDRESS	The string is of a net address.
ADSTYPE_REPLICAPOINTER	The data is of a replica pointer.
ADSTYPE_FAXNUMBER	The string is of a fax number.
ADSTYPE_EMAIL	The data is of an e-mail message.
ADSTYPE_NT_SECURITY_DESCRIPTOR	The data is of Windows NT/Windows 2000 Security Descriptor as represented by a byte array.
ADSTYPE_UNKNOWN	The data is of an undefined type.
ADSTYPE_DN_WITH_BINARY	The data is of ADS_DN_WITH_BINARY used for mapping a distinguished name to a non varying GUID.
ADSTYPE_DN_WITH_STRING	The data is of ADS_DN_WITH_STRING used for mapping a distinguished name to a non-varying string value.
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemAuditType ==

The EntityItemAuditType restricts a string value to a specific set of values: AUDIT_NONE, AUDIT_SUCCESS, AUDIT_FAILURE, and AUDIT_SUCCESS_FAILURE. These values describe which audit records should be generated. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 814: Enumeration Values

Value	Description
AUDIT_FAILURE	The audit type AUDIT_FAILURE is used to perform audits on all unsuccessful occurrences of specified events when auditing is enabled.
AUDIT_NONE	The audit type AUDIT_NONE is used to cancel all auditing options for the specified events.
AUDIT_SUCCESS	The audit type AUDIT_SUCCESS is used to perform audits on all successful occurrences of the specified events when auditing is enabled.
AUDIT_SUCCESS_FAILURE	The audit type AUDIT_SUCCESS_FAILURE is used to perform audits on all successful and unsuccessful occurrences of the specified events when auditing is enabled.
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemDriveTypeType ==

The EntityItemDriveTypeType complex type defines the different values that are valid for the drive_type entity of a win-sc:volume_item. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 815: Enumeration Values

Value	Description
DRIVE_UNKNOWN	The DRIVE_UNKNOWN type means that drive type cannot be determined. The UINT value that this corresponds to is 0.
DRIVE_NO_ROOT_DIR	The DRIVE_NO_ROOT_DIR type means that the root path is not valid. The UINT value that this corresponds to is 1.
DRIVE_REMOVABLE	The DRIVE_REMOVABLE type means that the drive contains removable media. The UINT value that this corresponds to is 2.
DRIVE_FIXED	The DRIVE_FIXED type means that the drive contains fixed media. The UINT value that this corresponds to is 3.
DRIVE_REMOTE	The DRIVE_REMOTE type means that the drive is a remote drive (i.e. network drive). The UINT value that this corresponds to is 4.
DRIVE_CDROM	The DRIVE_CDROM type means that the drive is a CD-ROM drive. The UINT value that this corresponds to is 5.
DRIVE_RAMDISK	The DRIVE_RAMDISK type means that the drive is a RAM disk. The UINT value that this corresponds to is 6.
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemFileTypeType ==

The EntityItemFileTypeType restricts a string value to a specific set of values that describe the different types of files. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 816: Enumeration Values

Value	Description
FILE_TYPE_CHAR	The specified file is a character file, typically an LPT device or a console.
FILE_TYPE_DISK	The specified file is a disk file.
FILE_TYPE_PIPE	The specified file is a socket, a named pipe, or an anonymous pipe.
FILE_TYPE_REMOTE	Unused.
FILE_TYPE_UNKNOWN	Either the type of the specified file is unknown, or the function failed.
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemFileAttributeType ==

The EntityItemFileAttributeType restricts a string value to a specific set of values that describe the different Windows file attributes. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 817: Enumeration Values

Value	Description
FILE_ATTRIBUTE_ARCHIVE	A file or directory that is an archive file or directory. Applications typically use this attribute to mark files for backup or removal.
FILE_ATTRIBUTE_COMPRESSED	A file or directory that is compressed. For a file, all of the data in the file is compressed. For a directory, compression is the default for newly created files and subdirectories.
FILE_ATTRIBUTE_DEVICE	This value is reserved for system use.
FILE_ATTRIBUTE_DIRECTORY	The handle that identifies a directory.
FILE_ATTRIBUTE_ENCRYPTED	A file or directory that is encrypted. For a file, all data streams in the file are encrypted. For a directory, encryption is the default for newly created files and subdirectories.
FILE_ATTRIBUTE_HIDDEN	The file or directory is hidden. It is not included in an ordinary directory listing.
FILE_ATTRIBUTE_INTEGRITY_STREAM	The directory or user data stream is configured with integrity (only supported on ReFS volumes). It is not included in an ordinary directory listing. The integrity setting persists with the file if it's renamed. If a file is copied the destination file will have integrity set if either the source file or destination directory have integrity set.Windows Server 2008 R2, Windows 7, Windows Server 2008, Windows Vista, Windows Server 2003, and Windows XP: This flag is not supported until Windows Server 2012.
FILE_ATTRIBUTE_NORMAL	A file that does not have other attributes set. This attribute is valid only when used alone.
FILE_ATTRIBUTE_NOT_CONTENT_INDEXED	The file or directory is not to be indexed by the content indexing service.
FILE_ATTRIBUTE_NO_SCRUB_DATA	The user data stream not to be read by the background data integrity scanner (AKA scrubber). When set on a directory it only provides inheritance. This flag is only

== EntityItemInterfaceTypeType ==

The EntityItemInterfaceTypeType restricts a string value to a specific set of values that describe the different types of interfaces. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 818: Enumeration Values

Value	Description
MIB_IF_TYPE_ETHERNET	The MIB_IF_TYPE_ETHERNET type is used to describe ethernet interfaces.
MIB_IF_TYPE_FDDI	The MIB_IF_TYPE_FDDI type is used to describe fiber distributed data interfaces (FDDI).
MIB_IF_TYPE_LOOPBACK	The MIB_IF_TYPE_LOOPBACK type is used to describe loopback interfaces.
MIB_IF_TYPE_OTHER	The MIB_IF_TYPE_OTHER type is used to describe unknown interfaces.
MIB_IF_TYPE_PPP	The MIB_IF_TYPE_PPP type is used to describe point-to-point protocol interfaces (PPP).
MIB_IF_TYPE_SLIP	The MIB_IF_TYPE_SLIP type is used to describe serial line internet protocol interfaces (SLIP).
MIB_IF_TYPE_TOKENRING	The MIB_IF_TYPE_TOKENRING type is used to describe token ring interfaces..
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemNamingContextType ==

The EntityItemNamingContextType restricts a string value to a specific set of values: domain, configuration, and schema. These values describe the different naming context found withing Active Directory. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 819: Enumeration Values

Value	Description
domain	The domain naming context contains Active Directory objects present in the specified domain (e.g. users, computers, groups, and other objects).
configuration	The configuration naming context contains configuration data that is required for the Active Directory to operate as a directory service.
schema	The schema naming context contains all of the Active Directory object definitions.
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemNTUserAccountTypeType ==

The EntityItemNTUserAccountTypeType restricts a string value to a specific set of values that describe the different types of accounts. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 820: Enumeration Values

Value	Description
local	Local accounts are accounts that were created directly on the machine being tested and should be in the form of machinenameusername
domain	Domain accounts are accounts that were created on a domain controller and should be in the form of domainusername
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemPeTargetMachineType ==

The EntityItemPeTargetMachineType enumeration identifies the valid machine targets that can be specified in the PE file header. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 821: Enumeration Values

Value	Description
IMAGE_FILE_MACHINE_UNKNOWN	The IMAGE_FILE_MACHINE_UNKNOWN type is used to indicate an unknown machine.
IMAGE_FILE_MACHINE_ALPHA	The IMAGE_FILE_MACHINE_ALPHA type is used to indicate an Alpha APX machine.
IMAGE_FILE_MACHINE_ARM	The IMAGE_FILE_MACHINE_ARM type is used to indicate an ARM little endian machine.
IMAGE_FILE_MACHINE_ALPHA64	The IMAGE_FILE_MACHINE_ALPHA64 type is used to indicate an 64-bit Alpha APX machine.
IMAGE_FILE_MACHINE_I386	The IMAGE_FILE_MACHINE_I386 type is used to indicate an Intel 386 machine.
IMAGE_FILE_MACHINE_IA64	The IMAGE_FILE_MACHINE_IA64 type is used to indicate an Intel Itanium machine.
IMAGE_FILE_MACHINE_M68K	The IMAGE_FILE_MACHINE_M68K type is used to indicate an M68K machine.
IMAGE_FILE_MACHINE_MIPS16	The IMAGE_FILE_MACHINE_MIPS16 type is used to indicate a MIPS16 machine.
IMAGE_FILE_MACHINE_MIPSFPU	The IMAGE_FILE_MACHINE_MIPSFPU type is used to indicate an MIPS machine with FPU.
IMAGE_FILE_MACHINE_MIPSFPU16	The IMAGE_FILE_MACHINE_MIPSFPU16 type is used to indicate a MIPS16 machine with FPU.
IMAGE_FILE_MACHINE_POWERPC	The IMAGE_FILE_MACHINE_POWERPC type is used to indicate an Power PC little endian machine.
IMAGE_FILE_MACHINE_R3000	The IMAGE_FILE_MACHINE_R3000 type is used to indicate a MIPS little endian, 0x160 big endian machine.
IMAGE_FILE_MACHINE_R4000	

== EntityItemPeSubsystemType ==

The EntityItemPeSubsystemType enumeration identifies the valid subsystem types that can be specified in the PE file header. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 822: Enumeration Values

Value	Description
IMAGE_SUBSYSTEM_UNKNOWN	The IMAGE_SUBSYSTEM_UNKNOWN type is used to indicate an unknown subsystem.
IMAGE_SUBSYSTEM_NATIVE	The IMAGE_SUBSYSTEM_NATIVE type is used to indicate that no subsystem is required.
IMAGE_SUBSYSTEM_WINDOWS_GUI	The IMAGE_SUBSYSTEM_WINDOWS_GUI type is used to indicate a Windows graphical user interface (GUI) subsystem.
IMAGE_SUBSYSTEM_WINDOWS_CUI	The IMAGE_SUBSYSTEM_WINDOWS_CUI type is used to indicate a Windows character-mode user interface (CUI) subsystem.
IMAGE_SUBSYSTEM_OS2_CUI	The IMAGE_SUBSYSTEM_OS2_CUI type is used to indicate an OS/2 CUI subsystem.
IMAGE_SUBSYSTEM_POSIX_CUI	The IMAGE_SUBSYSTEM_POSIX_CUI type is used to indicate a POSIX CUI subsystem.
IMAGE_SUBSYSTEM_WINDOWS_CE_GUI	The IMAGE_SUBSYSTEM_WINDOWS_CE_GUI type is used to indicate a Windows CE system.
IMAGE_SUBSYSTEM_EFI_APPLICATION	The IMAGE_SUBSYSTEM_EFI_APPLICATION type is used to indicate an Extensible Firmware Interface (EFI) application.
IMAGE_SUBSYSTEM_EFI_BOOT_SERVICE_DRIVER	The IMAGE_SUBSYSTEM_EFI_BOOT_SERVICE_DRIVER type is used to indicate a EFI driver with boot services.
IMAGE_SUBSYSTEM_EFI_RUNTIME_DRIVER	The IMAGE_SUBSYSTEM_EFI_RUNTIME_DRIVER type is used to indicate a EFI driver with run-time services subsystem.
IMAGE_SUBSYSTEM_EFI_ROM	The IMAGE_SUBSYSTEM_EFI_ROM type is used to indicate an EFI ROM image.
IMAGE_SUBSYSTEM_XBOX	

== EntityItemProtocolType ==

The EntityItemProtocolType restricts a string value to a specific set of values that describe the different available protocols. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 823: Enumeration Values

Value	Description
TCP	The port uses the Transmission Control Protocol (TCP).
UDP	The port uses the User Datagram Protocol (UDP).
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemRegistryHiveType ==

The EntityItemRegistryHiveType restricts a string value to a specific set of values that describe the different registry hives. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 824: Enumeration Values

Value	Description
HKEY_CLASSES_ROOT	This registry subtree contains information that associates file types with programs and configuration data for automation (e.g. COM objects and Visual Basic Programs).
HKEY_CURRENT_CONFIG	This registry subtree contains configuration data for the current hardware profile.
HKEY_CURRENT_USER	This registry subtree contains the user profile of the user that is currently logged into the system.
HKEY_CURRENT_USER_LOCAL_SETTINGS	Registry entries subordinate to this key define preferences of the current user that are local to the machine. These entries are not included in the per-user registry portion of a roaming user profile. This key is supported starting with Windows 7 and Windows Server 2008 R2.
HKEY_LOCAL_MACHINE	This registry subtree contains information about the local system.
HKEY_USERS	This registry subtree contains user-specific data.
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemRegistryTypeType ==

The EntityItemRegistryTypeType defines the different values that are valid for the type entity of a registry item. These values describe the possible types of data stored in a registry key. restricts a string value to a specific set of values that describe the different registry types. The empty string is also allowed as a valid value to support empty emlements associated with error conditions. Please note that the values identified are for the type entity and are not valid values for the datatype attribute. For information about how to encode registry data in OVAL for each of the different types, please visit the registry_item documentation.

Restricts: oval-sc:EntityItemStringType

Table 825: Enumeration Values

Value	Description
reg_binary	The reg_binary type is used by registry keys that specify binary data in any form.
reg_dword	The reg_dword type is used by registry keys that specify an unsigned 32-bit integer.
reg_dword_little_endian (Deprecated)	The reg_dword_little_endian type is used by registry keys that specify an unsigned 32-bit little-endian integer. It is designed to run on little-endian computer architectures. **Deprecated As Of Version:** 5.11.1:1.1 **Reason:** Defined to have same value as reg_dword. **Comment:** This registry type enumeration value has been deprecated and may be removed in a future version of the language.
reg_dword_big_endian	The reg_dword_big_endian type is used by registry keys that specify an unsigned 32-bit big-endian integer. It is designed to run on big-endian computer architectures.
reg_expand_sz	The reg_expand_sz type is used by registry keys to specify a null-terminated string that contains unexpanded references to environment variables (for example, “%PATH%”).
reg_link	The reg_link type is used by the registry keys for null-terminated unicode strings. It is related to target path of a symbolic link created by the RegCreateKeyEx function.
reg_multi_sz	The reg_multi_sz type is used by registry keys that specify an array of null-terminated strings, terminated by two null characters.
reg_none	The reg_none type is used by registry keys that have no defined value type.
reg_qword	The reg_qword type is used by registry keys that specify an unsigned 64-bit integer.
reg_qword_little_endian (Deprecated)	

== EntityItemServiceControlsAcceptedType ==

The EntityItemServiceAcceptedControlsType complex type defines the different values that are valid for the controls_accepted entity of a service. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 826: Enumeration Values

Value	Description
SERVICE_ACCEPT_NETBINDCHANGE	The SERVICE_ACCEPT_NETBINDCHANGE type means that the service is a network component and can accept changes in its binding without being stopped or restarted. The DWORD value that this corresponds to is 0x00000010.
SERVICE_ACCEPT_PARAMCHANGE	The SERVICE_ACCEPT_PARAMCHANGE type means that the service can re-read its startup parameters without being stopped or restarted. The DWORD value that this corresponds to is 0x00000008.
SERVICE_ACCEPT_PAUSE_CONTINUE	The SERVICE_ACCEPT_PAUSE_CONTINUE type means that the service can be paused or continued. The DWORD value that this corresponds to is 0x00000002.
SERVICE_ACCEPT_PRESHUTDOWN	The SERVICE_ACCEPT_PRESHUTDOWN type means that the service can receive pre-shutdown notifications. The DWORD value that this corresponds to is 0x00000100.
SERVICE_ACCEPT_SHUTDOWN	The SERVICE_ACCEPT_SHUTDOWN type means that the service can receive shutdown notifications. The DWORD value that this corresponds to is 0x00000004.
SERVICE_ACCEPT_STOP	The SERVICE_ACCEPT_STOP type means that the service can be stopped. The DWORD value that this corresponds to is 0x00000001.
SERVICE_ACCEPT_HARDWAREPROFILECHANGE	The SERVICE_ACCEPT_HARDWAREPROFILECHANGE type means that the service can receive notifications when the system's hardware profile changes. The DWORD value that this corresponds to is 0x00000020.
SERVICE_ACCEPT_POWEREVENT	The SERVICE_ACCEPT_POWEREVENT type means that the service can receive notifications when the system's power status has changed. The DWORD value that this corresponds to is 0x00000040.
SERVICE_ACCEPT_SESSIONCHANGE	The SERVICE_ACCEPT_SESSIONCHANGE type means that the service can receive notifications when the system's session status has changed. The DWORD value that this corresponds to is 0x00000080.

== EntityItemServiceCurrentStateType ==

The EntityItemServiceCurrentStateType complex type defines the different values that are valid for the current_state entity of a service. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 827: Enumeration Values

Value	Description
SERVICE_CONTINUE_PENDING	The SERVICE_CONTINUE_PENDING type means that the service has been sent a command to continue, however, the command has not yet been executed. The DWORD value that this corresponds to is 0x00000005.
SERVICE_PAUSE_PENDING	The SERVICE_PAUSE_PENDING type means that the service has been sent a command to pause, however, the command has not yet been executed. The DWORD value that this corresponds to is 0x00000006.
SERVICE_PAUSED	The SERVICE_PAUSED type means that the service is paused. The DWORD value that this corresponds to is 0x00000007.
SERVICE_RUNNING	The SERVICE_RUNNING type means that the service is running. The DWORD value that this corresponds to is 0x00000004.
SERVICE_START_PENDING	The SERVICE_START_PENDING type means that the service has been sent a command to start, however, the command has not yet been executed. The DWORD value that this corresponds to is 0x00000002.
SERVICE_STOP_PENDING	The SERVICE_STOP_PENDING type means that the service has been sent a command to stop, however, the command has not yet been executed. The DWORD value that this corresponds to is 0x00000003.
SERVICE_STOPPED	The SERVICE_STOPPED type means that the service is stopped. The DWORD value that this corresponds to is 0x00000001.
	The empty string value is permitted here to allow for empty elements associated with error conditions.

== EntityItemServiceStartTypeType ==

The EntityItemServiceStartTypeType complex type defines the different values that are valid for the start_type entity of a service. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 828: Enumeration Values

Value	Description
SERVICE_AUTO_START	The SERVICE_AUTO_START type means that the service is started automatically by the Service Control Manager (SCM) during startup. The DWORD value that this corresponds to is 0x00000002.
SERVICE_BOOT_START	The SERVICE_BOOT_START type means that the driver service is started by the system loader. The DWORD value that this corresponds to is 0x00000000.
SERVICE_DEMAND_START	The SERVICE_DEMAND_START type means that the service is started by the Service Control Manager (SCM) when StartService() is called. The DWORD value that this corresponds to is 0x00000003.
SERVICE_DISABLED	The SERVICE_DISABLED type means that the service cannot be started. The DWORD value that this corresponds to is 0x00000004.
SERVICE_SYSTEM_START	The SERVICE_SYSTEM_START type means that the service is a device driver started by IoInitSystem(). The DWORD value that this corresponds to is 0x00000001.
	The empty string value is permitted here to allow for empty elements associated with error conditions.

== EntityItemServiceTypeType ==

The EntityItemServiceTypeType complex type defines the different values that are valid for the service_type entity of a service. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 829: Enumeration Values

Value	Description
SERVICE_FILE_SYSTEM_DRIVER	The SERVICE_FILE_SYSTEM_DRIVER type means that the service is a file system driver. The DWORD value that this corresponds to is 0x00000002.
SERVICE_KERNEL_DRIVER	The SERVICE_KERNEL_DRIVER type means that the service is a driver. The DWORD value that this corresponds to is 0x00000001.
SERVICE_WIN32_OWN_PROCESS	The SERVICE_WIN32_OWN_PROCESS type means that the service runs in its own process. The DWORD value that this corresponds to is 0x00000010.
SERVICE_WIN32_SHARE_PROCESS	The SERVICE_WIN32_SHARE_PROCESS type means that the service runs in a process with other services. The DWORD value that this corresponds to is 0x00000020.
SERVICE_INTERACTIVE_PROCESS	The SERVICE_WIN32_SHARE_PROCESS type means that the service runs in a process with other services. The DWORD value that this corresponds to is 0x00000100.
	The empty string value is permitted here to allow for empty elements associated with error conditions.

== EntityItemSharedResourceTypeType ==

The EntityItemSharedResourceTypeType complex type defines the different values that are valid for the type entity of a shared resource item. Note that the Windows API returns a DWORD value and OVAL uses the constant name that is normally defined for these return values. This is done to increase readability and maintainability of OVAL Definitions. The empty string is also allowed to support empty elements associated with error conditions.

It is also important to note that special shared resources are those reserved for remote administration, interprocess communication, and administrative shares.

Restricts: oval-sc:EntityItemStringType

Table 830: Enumeration Values

Value	Description
STYPE_DISKTREE	The STYPE_DISKTREE type means that the shared resource is a disk drive. The DWORD value that this corresponds to is 0x00000000.
STYPE_DISKTREE_SPECIAL	The STYPE_DISKTREE_SPECIAL type means that the shared resource is a special disk drive. The DWORD value that this corresponds to is 0x80000000.
STYPE_DISKTREE_TEMPORARY	The STYPE_DISKTREE_TEMPORARY type means that the shared resource is a temporary disk drive. The DWORD value that this corresponds to is 0x40000000.
STYPE_DISKTREE_SPECIAL_TEMPORARY	The STYPE_DISKTREE_SPECIAL_TEMPORARY type means that the shared resource is a temporary, special disk drive. The DWORD value that this corresponds to is 0xC0000000.
STYPE_PRINTQ	The STYPE_PRINTQ type means that the shared resource is a print queue. The DWORD value that this corresponds to is 0x00000001.
STYPE_PRINTQ_SPECIAL	The STYPE_PRINTQ_SPECIAL type means that the shared resource is a special print queue. The DWORD value that this corresponds to is 0x80000001.
STYPE_PRINTQ_TEMPORARY	The STYPE_PRINTQ_TEMPORARY type means that the shared resource is a temporary print queue. The DWORD value that this corresponds to is 0x40000001.
STYPE_PRINTQ_SPECIAL_TEMPORARY	The STYPE_PRINTQ_SPECIAL_TEMPORARY type means that the shared resource is a temporary, special print queue. The DWORD value that this corresponds to is 0xC0000001.
STYPE_DEVICE	The STYPE_DEVICE type means that the shared resource is a communication device. The DWORD value that this corresponds to is 0x00000002.
STYPE_DEVICE_SPECIAL	The STYPE_DEVICE_SPECIAL type means that the shared resource is a special communication device. The DWORD value that this corresponds to is 0x80000002.

== EntityItemSystemMetricIndexType ==

The EntityItemSystemMetricIndexType complex type defines the different values that are valid for the index entity of a system_metric item. These values describe the system metric or configuration setting to be retrieved. The empty string is also allowed to support empty elements associated with error conditions. Please note that the values identified are for the index entity and are not valid values for the datatype attribute.

Restricts: oval-sc:EntityItemStringType

Table 831: Enumeration Values

Value	Description
SM_ARRANGE	The flags that specify how the system arranged minimized windows.
SM_CLEANBOOT	The value that specifies how the system is started.
SM_CMONITORS	The number of display monitors on a desktop.
SM_CMOUSEBUTTONS	The number of buttons on a mouse, or zero if no mouse is installed.
SM_CXBORDER	The width of a window border, in pixels. This is equivalent to the SM_CXEDGE value for windows with the 3-D look.
SM_CXCURSOR	The width of a cursor, in pixels. The system cannot create cursors of other sizes.
SM_CXDLGFRAME	This value is the same as SM_CXFIXEDFRAME.
SM_CXDOUBLECLK	The width of the rectangle around the location of a first click in a double-click sequence, in pixels.
SM_CXDRAG	The number of pixels on either side of a mouse-down point that the mouse pointer can move before a drag operation begins.
SM_CXEDGE	The width of a 3-D border, in pixels. This metric is the 3-D counterpart of SM_CXBORDER.

Continued on next page

Table 831 – continued from previous page

Value	Description
SM_CXFIXEDFRAME	The thickness of the frame around the perimeter of a window that has a caption but is not sizable, in pixels.
SM_CXFOCUSBORDER	The width of the left and right edges of the focus rectangle that the DrawFocusRect draws.
SM_CXFRAME	This value is the same as SM_CXSIZEFRAME.
SM_CXFULLSCREEN	The width of the client area for a full-screen window on the primary display monitor, in pixels.
SM_CXHSCROLL	The width of the arrow bitmap on a horizontal scroll bar, in pixels.
SM_CXHTHUMB	The width of the thumb box in a horizontal scroll bar, in pixels.
SM_CXICON	The default width of an icon, in pixels.
SM_CXICONSPACING	The width of a grid cell for items in large icon view, in pixels.
SM_CXMAXIMIZED	The default width, in pixels, of a maximized top-level window on the primary display monitor.
SM_CXMAXTRACK	The default maximum width of a window that has a caption and sizing borders, in pixels.
SM_CXMENUCHECK	The width of the default menu check-mark bitmap, in pixels.
SM_CXMENUSIZE	The width of menu bar buttons, such as the child window close button that is used in the multiple document interface, in pixels.
SM_CXMIN	The minimum width of a window, in pixels.

Continued on next page

Table 831 – continued from previous page

Value	Description
SM_CXMINIMIZED	The width of a minimized window, in pixels.
SM_CXMINSPACING	The width of a grid cell for a minimized window, in pixels.
SM_CXMINTRACK	The minimum tracking width of a window, in pixels.
SM_CXPADDEDBORDER	The amount of border padding for captioned windows, in pixels.
SM_CXSCREEN	The width of the screen of the primary display monitor, in pixels.
SM_CXSIZE	The width of a button in a window caption or title bar, in pixels.
SM_CXSIZEFRAME	The thickness of the sizing border around the perimeter of a window that can be resized, in pixels.
SM_CXSMICON	The recommended width of a small icon, in pixels.
SM_CXSMSIZE	The width of small caption buttons, in pixels.
SM_CXVIRTUALSCREEN	The width of the virtual screen, in pixels.
SM_CXVSCROLL	The width of a vertical scroll bar, in pixels.
SM_CYBORDER	The height of a window border, in pixels.
SM_CYCAPTION	The height of a caption area, in pixels.
SM_CYCURSOR	The height of a cursor, in pixels.
SM_CYDLGFRAME	This value is the same as SM_CYFIXEDFRAME.

Continued on next page

Table 831 – continued from previous page

Value	Description
SM_CYDOUBLECLK	The height of the rectangle around the location of a first click in a double-click sequence, in pixels.
SM_CYDRAG	The number of pixels above and below a mouse-down point that the mouse pointer can move before a drag operation begins.
SM_CYEDGE	The height of a 3-D border, in pixels. This is the 3-D counterpart of SM_CYBORDER.
SM_CYFIXEDFRAME	The thickness of the frame around the perimeter of a window that has a caption but is not sizable, in pixels.
SM_CYFOCUSBORDER	The height of the top and bottom edges of the focus rectangle drawn by DrawFocusRect. This value is in pixels.
SM_CYFRAME	This value is the same as SM_CYSIZEFRAME.
SM_CYFULLSCREEN	The height of the client area for a full-screen window on the primary display monitor, in pixels.
SM_CYHSCROLL	The height of a horizontal scroll bar, in pixels.
SM_CYICON	The default height of an icon, in pixels.
SM_CYICONSPACING	The height of a grid cell for items in large icon view, in pixels.
SM_CYKANJIWINDOW	For double byte character set versions of the system, this is the height of the Kanji window at the bottom of the screen, in pixels.
SM_CYMAXIMIZED	The default height, in pixels, of a maximized top-level window on the primary display monitor.

Continued on next page

Table 831 – continued from previous page

Value	Description
SM_CYMAXTRACK	The default maximum height of a window that has a caption and sizing borders, in pixels.
SM_CYMENU	The height of a single-line menu bar, in pixels.
SM_CYMENUCHECK	The height of the default menu check-mark bitmap, in pixels.
SM_CYMENUSIZE	The height of menu bar buttons, such as the child window close button that is used in the multiple document interface, in pixels.
SM_CYMIN	The minimum height of a window, in pixels.
SM_CYMINIMIZED	The height of a minimized window, in pixels.
SM_CYMINSPACING	The height of a grid cell for a minimized window, in pixels.
SM_CYMINTRACK	The minimum tracking height of a window, in pixels.
SM_CYSCREEN	The height of the screen of the primary display monitor, in pixels.
SM_CYSIZE	The height of a button in a window caption or title bar, in pixels.
SM_CYSIZEFRAME	The thickness of the sizing border around the perimeter of a window that can be resized, in pixels.
SM_CYSMCAPTION	The height of a small caption, in pixels.
SM_CYSMICON	The recommended height of a small icon, in pixels.
SM_CYSMSIZE	The height of small caption buttons, in pixels.

Continued on next page

Table 831 – continued from previous page

Value	Description
SM_CYVIRTUALSCREEN	The height of the virtual screen, in pixels. The virtual screen is the bounding rectangle of all display monitors.
SM_CYVSCROLL	The height of the arrow bitmap on a vertical scroll bar, in pixels.
SM_CYVTHUMB	The height of the thumb box in a vertical scroll bar, in pixels.
SM_DBCSENABLED	Nonzero if User32.dll supports DBCS; otherwise, 0.
SM_DEBUG	Nonzero if the debug version of User.exe is installed; otherwise, 0.
SM_DIGITIZER	Nonzero if the current operating system is Windows 7 or Windows Server 2008 R2 and the Tablet PC Input service is started; otherwise, 0. The return value is a bitmask that specifies the type of digitizer input supported by the device.
SM_IMMENABLED	Nonzero if Input Method Manager/Input Method Editor features are enabled; otherwise, 0.
SM_MAXIMUMTOUCHES	Nonzero if there are digitizers in the system; otherwise, 0.
SM_MEDIACENTER	Nonzero if the current operating system is the Windows XP, Media Center Edition, 0 if not.
SM_MENUDROPALIGNMENT	Nonzero if drop-down menus are right-aligned with the corresponding menu-bar item; 0 if the menus are left-aligned.
SM_MIDEASTENABLED	Nonzero if the system is enabled for Hebrew and Arabic languages, 0 if not.
SM_MOUSEPRESENT	Nonzero if a mouse is installed; otherwise, 0.

Continued on next page

Table 831 – continued from previous page

Value	Description
SM_MOUSEHORIZONTALWHEELPRESENT	Nonzero if a mouse with a horizontal scroll wheel is installed; otherwise 0.
SM_MOUSEWHEELPRESENT	Nonzero if a mouse with a vertical scroll wheel is installed; otherwise 0.
SM_NETWORK	The least significant bit is set if a network is present; otherwise, it is cleared.
SM_PENWINDOWS	Nonzero if the Microsoft Windows for Pen computing extensions are installed; zero otherwise.
SM_REMOTECONTROL	This system metric is used in a Terminal Services environment to determine if the current Terminal Server session is being remotely controlled. Its value is nonzero if the current session is remotely controlled; otherwise, 0.
SM_REMOTESESSION	This system metric is used in a Terminal Services environment. If the calling process is associated with a Terminal Services client session, the return value is nonzero. If the calling process is associated with the Terminal Services console session, the return value is 0.
SM_SAMEDISPLAYFORMAT	Nonzero if all the display monitors have the same color format, otherwise, 0.
SM_SECURE	This system metric should be ignored; it always returns 0.
SM_SERVERR2	The build number if the system is Windows Server 2003 R2; otherwise, 0.
SM_SHOWSOUNDS	Nonzero if the user requires an application to present information visually in situations where it would otherwise present the information only in audible form; otherwise, 0.

Continued on next page

Table 831 – continued from previous page

Value	Description
SM_SHUTTINGDOWN	Nonzero if the current session is shutting down; otherwise, 0.
SM_SLOWMACHINE	Nonzero if the computer has a low-end (slow) processor; otherwise, 0.
SM_STARTER	Nonzero if the current operating system is Windows 7 Starter Edition, Windows Vista Starter, or Windows XP Starter Edition; otherwise, 0.
SM_SWAPBUTTON	Nonzero if the meanings of the left and right mouse buttons are swapped; otherwise, 0.
SM_TABLETPC	Nonzero if the current operating system is the Windows XP Tablet PC edition or if the current operating system is Windows Vista or Windows 7 and the Tablet PC Input service is started; otherwise, 0.
SM_XVIRTUALSCREEN	The coordinates for the left side of the virtual screen.
SM_YVIRTUALSCREEN	The coordinates for the top of the virtual screen.
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemGUIDType ==

The EntityItemGUIDType restricts a string value to a representation of a GUID, used for module ID. The empty string is also allowed to support empty element associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Pattern: ({[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{12}}){0,}

== EntityItemCmdletVerbType ==

The EntityItemCmdletVerbType restricts a string value to a set of allow cmdlet verbs. The empty string is also allowed to support empty element associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 832: Enumeration Values

Value	Description
Approve	The Approve verb confirms or agrees to the status of a resource or process.
Assert	The Assert verb affirms the state of a resource.
Compare	The Compare verb evaluates the data from one resource against the data from another resource.
Confirm	The Confirm verb acknowledges, verifies, or validates, the state of a resource or process.
Find	The Find verb looks for an object in a container that is unknown, implied, optional, or specified.
Get	The Get verb specifies an action that retrieves a resource.
Import	The Import verb creates a resource from data that is stored in a persistent data store (such as a file) or in an interchange format.
Measure	The Measure verb identifies resources that are consumed by a specified operation, or retrieves statistics about a resource.
Read	The Read verb acquires information from a source.
Request	The Request verb asks for a resource or asks for permissions.
Resolve	The Resolve verb maps a shorthand representation of a resource to a more complete representation.
Search	The Search verb creates a reference to a resource in a container.
Select	The Select verb locates a resource in a container

== EntityItemWindowsViewType ==

The EntityItemWindowsViewType restricts a string value to a specific set of values: 32-bit and 64-bit. These values describe the different values possible for the windows view behavior.

Restricts: oval-sc:EntityItemStringType

Table 833: Enumeration Values

Value	Description
32_bit	Indicates the 32_bit windows view.
64_bit	Indicates the 64_bit windows view.
	The empty string value is permitted here to allow for empty elements associated with error conditions.

== EntityItemUserRightType ==

The EntityItemUserRightType restricts a string value to a specific set of values that describe the different user rights/privileges. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 834: Enumeration Values

Value	Description
SE_ASSIGNPRIMARYTOKEN_NAME	This privilege is required to assign the primary token of a process.
SE_AUDIT_NAME	This privilege is required to generate audit-log entries.
SE_BACKUP_NAME	This privilege is required to perform backup operations.
SE_CHANGE_NOTIFY_NAME	This privilege is required to receive notifications of changes to files or directories.
SE_CREATE_GLOBAL_NAME	This privilege is required to create named file mapping objects in the global namespace during Terminal Services sessions.

Continued on next page

Table 834 – continued from previous page

Value	Description
SE_CREATE_PAGEFILE_NAME	This privilege is required to create a paging file.
SE_CREATE_PERMANENT_NAME	This privilege is required to create a permanent object.
SE_CREATE_SYMBOLIC_LINK_NAME	This privilege is required to create a symbolic link.
SE_CREATE_TOKEN_NAME	This privilege is required to create a primary token.
SE_DEBUG_NAME	This privilege is required to debug and adjust the memory of a process owned by another account.
SE_ENABLE_DELEGATION_NAME	This privilege is required to mark user and computer accounts as trusted for delegation.
SE_IMPERSONATE_NAME	This privilege is required to impersonate.
SE_INC_BASE_PRIORITY_NAME	This privilege is required to increase the base priority of a process.
SE_INCREASE_QUOTA_NAME	This privilege is required to increase the quota assigned to a process.
SE_INC_WORKING_SET_NAME	This privilege is required to allocate more memory for applications that run in the context of users.
SE_LOAD_DRIVER_NAME	This privilege is required to load or unload a device driver.
SE_LOCK_MEMORY_NAME	This privilege is required to lock physical pages in memory.
SE_MACHINE_ACCOUNT_NAME	This privilege is required to create a computer account.
SE_MANAGE_VOLUME_NAME	This privilege is required to enable volume management privileges.

Continued on next page

Table 834 – continued from previous page

Value	Description
SE_PROF_SINGLE_PROCESS_NAME	This privilege is required to gather profiling information for a single process.
SE_RELABEL_NAME	This privilege is required to modify the mandatory integrity level of an object.
SE_REMOTE_SHUTDOWN_NAME	This privilege is required to shut down a system using a network request.
SE_RESTORE_NAME	This privilege is required to perform restore operations.
SE_SECURITY_NAME	This privilege is required to perform a number of security-related functions, such as controlling and viewing audit messages.
SE_SHUTDOWN_NAME	This privilege is required to shut down a local system.
SE_SYNC_AGENT_NAME	This privilege is required for a domain controller to use the Lightweight Directory Access Protocol directory synchronization services.
SE_SYSTEM_ENVIRONMENT_NAME	This privilege is required to modify the nonvolatile RAM of systems that use this type of memory to store configuration information.
SE_SYSTEM_PROFILE_NAME	This privilege is required to gather profiling information for the entire system.
SE_SYSTEMTIME_NAME	This privilege is required to modify the system time.
SE_TAKE_OWNERSHIP_NAME	This privilege is required to take ownership of an object without being granted discretionary access.
SE_TCB_NAME	This privilege identifies its holder as part of the trusted computer base.

Continued on next page

Table 834 – continued from previous page

Value	Description
SE_TIME_ZONE_NAME	This privilege is required to adjust the time zone associated with the computer's internal clock.
SE_TRUSTED_CREDMAN_ACCESS_NAME	This privilege is required to access Credential Manager as a trusted caller.
SE_UNDOCK_NAME	This privilege is required to undock a laptop.
SE_UNSOLICITED_INPUT_NAME	This privilege is required to read unsolicited input from a terminal device.
SE_BATCH_LOGON_NAME	This account right is required for an account to log on using the batch logon type.
SE_DENY_BATCH_LOGON_NAME	This account right explicitly denies an account the right to log on using the batch logon type.
SE_DENY_INTERACTIVE_LOGON_NAME	This account right explicitly denies an account the right to log on using the interactive logon type.
SE_DENY_NETWORK_LOGON_NAME	This account right explicitly denies an account the right to log on using the network logon type.
SE_DENY_REMOTE_INTERACTIVE_LOGON_NAME	This account right explicitly denies an account the right to log on remotely using the interactive logon type.
SE_DENY_SERVICE_LOGON_NAME	This account right explicitly denies an account the right to log on using the service logon type.
SE_INTERACTIVE_LOGON_NAME	This account right is required for an account to log on using the interactive logon type.
SE_NETWORK_LOGON_NAME	This account right is required for an account to log on using the network logon type.

Continued on next page

Table 834 – continued from previous page

Value	Description
SE_REMOTE_INTERACTIVE_LOGON_NAME	This account right is required for an account to log on remotely using the interactive logon type.
SE_SERVICE_LOGON_NAME	This account right is required for an account to log on using the service logon type.
	The empty string value is permitted here to allow for detailed error reporting.

Open Vulnerability and Assessment Language: SharePoint Definition

- Schema: SharePoint Definition
- Version: 5.11.1:1.1
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the SharePoint specific tests found in Open Vulnerability and Assessment Language (OVAL). Each test is an extension of the standard test element defined in the Core Definition Schema. Through extension, each test inherits a set of elements and attributes that are shared amongst all OVAL tests. Each test is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core Definition Schema is not outlined here.

The SharePoint Component Schema is based on the SharePoint Object Model (Windows SharePoint Services 3.0)

The OVAL Schema is maintained by the 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.cisecurity.org.

Test Listing

- *< spwebapplication_test >*
- *< spgroup_test >*
- *< spweb_test >*
- *< splist_test >*
- *< spantivirussettings_test >*
- *< spsiteadministration_test >*
- *< spsite_test >*
- *< spcrawlrule_test >*
- *< spjobdefinition_test > (Deprecated)* (Deprecated)
- *< spjobdefinition510_test >*
- *< bestbet_test >*

- *< infopolicycoll_test >*
- *< spdiagnosticsservice_test >*
- *< spdiagnosticslevel_test >*
- *< sppolicyfeature_test >*
- *< sppolicy_test >*

< spwebapplication_test >

The spwebapplication test is used to check the properties or permission settings of a SharePoint web application. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a spwebapplication_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 835: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..1)	

< spwebapplication_object >

The spwebapplication_object element is used by a spwebapplication test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An spwebapplication object consists of a webapplicationurl used to define a specific web application. See the defintion of the SPWebApplication class in the SharePoint object model documentation.

Extends: oval-def:ObjectType

Child Elements

Table 836: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
webapplicationurl	oval-def:EntityObjectStringType (1..1)	The webapplicationurl element defines the SPWebApplication to evaluate specific security settings or permissions.
oval-def:filter	n/a (0..unbounded)	

< spwebapplication_state >

The spwebapplication_state element defines security settings and permissions that can be checked for a specified SPWebApplications.

Extends: oval-def:StateType

Child Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
webapplicationurl	oval-def:EntityStateStringType (0..1)	The webapplicationurl element identifies a Web a
allowparttopartcommunication	oval-def:EntityStateBoolType (0..1)	If the allowparttopartcommunication is enabled it
allowaccesstowebpartcatalog	oval-def:EntityStateBoolType (0..1)	If the allowaccesstowebpartcatalog is enabled it a
blockedfileextention	oval-def:EntityStateStringType (0..1)	The blockedfileextention element identifies one o
defaultquotatemplate	oval-def:EntityStateStringType (0..1)	The defaultquotatemplate element identifies the d
externalworkflowparticipantsenabled	oval-def:EntityStateBoolType (0..1)	If the externalworkflowparticipantsenabled is ena
recyclebinenabled	oval-def:EntityStateBoolType (0..1)	If the recyclebinenabled is enabled it will be easy
automaticallydeleteunusedsitecollections	oval-def:EntityStateBoolType (0..1)	If the automaticallydeleteunusedsitecollections is
selfservicesitecreationenabled	oval-def:EntityStateBoolType (0..1)	If the selfservicesitecreationenabled is enabled us
secondstagerecyclebinquota	oval-def:EntityStateIntType (0..1)	The secondstagerecyclebinquota is the quota for t
recyclebinretentionperiod	oval-def:EntityStateIntType (0..1)	The recyclebinretentionperiod is the retention per
outboundmailserverinstance	oval-def:EntityStateStringType (0..1)	The outboundmailserverinstance element identifie
outboundmailsenderaddress	oval-def:EntityStateStringType (0..1)	The outboundmailsenderaddress element identifie
outboundmailreplytoaddress	oval-def:EntityStateStringType (0..1)	The outboundmailreplytoaddress element identifi
secvalexpires	oval-def:EntityStateBoolType (0..1)	If the secvalexpires is enabled then the form will
timeout	oval-def:EntityStateIntType (0..1)	The timeout is the amount of time before security
isadministrationwebapplication	oval-def:EntityStateBoolType (0..1)	If this is true, the web application to which this te
applicationpoolname	oval-def:EntityStateStringType (0..1)	The applicationpoolname element identifies the w
applicationpoolusername	oval-def:EntityStateStringType (0..1)	The applicationpoolusername element identifies t
openitems	oval-def:EntityStateBoolType (0..1)	If the openitems is enabled the permission to view
addlistitems	oval-def:EntityStateBoolType (0..1)	If the addlistitems is enabled the permission to ad
approveitems	oval-def:EntityStateBoolType (0..1)	If approveitems is enabled the permission to appr
deletelistitems	oval-def:EntityStateBoolType (0..1)	If the deletelistitems is enabled the permission to
deleteversions	oval-def:EntityStateBoolType (0..1)	If the deleteversions is enabled the permission to
editlistitems	oval-def:EntityStateBoolType (0..1)	If the editlistitems is enabled the permission to ed
managelists	oval-def:EntityStateBoolType (0..1)	If the managelists is enabled the permission to cre
viewversions	oval-def:EntityStateBoolType (0..1)	If the viewversions is enabled the permission to v
viewlistitems	oval-def:EntityStateBoolType (0..1)	If the viewlistitems is enabled the permission to v
cancelcheckout	oval-def:EntityStateBoolType (0..1)	If the cancelcheckout is enabled the permission to
createalerts	oval-def:EntityStateBoolType (0..1)	If the createalerts is enabled the permission to Cre
viewformpages	oval-def:EntityStateBoolType (0..1)	If the viewformpages is enabled the permission to
viewpages	oval-def:EntityStateBoolType (0..1)	If the viewpages is enabled the permission to view
addandcustomizepages	oval-def:EntityStateBoolType (0..1)	If addandcustomizepages is enabled the permissio
applystylesheets	oval-def:EntityStateBoolType (0..1)	If the applystylesheets is enabled the permission t
applythemeandborder	oval-def:EntityStateBoolType (0..1)	If the applythemeanborder is enabled the permiss
browsedirectories	oval-def:EntityStateBoolType (0..1)	If the browsedirectories is enabled the permission
browseuserinfo	oval-def:EntityStateBoolType (0..1)	If the browseuserinfo is enabled the permission to
creategroups	oval-def:EntityStateBoolType (0..1)	If the creategroups is enabled the permission to cr
createsscsite	oval-def:EntityStateBoolType (0..1)	If the createsscsite is enabled the permission to cr

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
editmyuserinfo	oval-def:EntityStateBoolType (0..1)	If the editmyuserinfo is enabled the permission to
enumeratepermissions	oval-def:EntityStateBoolType (0..1)	If enumeratepermissions is enabled the permissio
managealerts	oval-def:EntityStateBoolType (0..1)	If the managealerts is enabled the permission to m
managepermissions	oval-def:EntityStateBoolType (0..1)	If the managepermissions is enabled the permissio
managesubwebs	oval-def:EntityStateBoolType (0..1)	If the managesubwebs is enabled the permission t
manageweb	oval-def:EntityStateBoolType (0..1)	If the manageweb is enabled the permission to pe
open	oval-def:EntityStateBoolType (0..1)	If open is enabled the permission to allow users to
useclientintegration	oval-def:EntityStateBoolType (0..1)	If the useclientintegration is enabled the permissio
useremoteapis	oval-def:EntityStateBoolType (0..1)	If the useremoteapis is enabled the permission to
viewusagedata	oval-def:EntityStateBoolType (0..1)	If the viewusagedata is enabled the permission to
managepersonalviews	oval-def:EntityStateBoolType (0..1)	If the managepersonalviews is enabled the permis
adddelprivatewebparts	oval-def:EntityStateBoolType (0..1)	If the adddelprivatewebparts is enabled the permis
updatepersonalwebparts	oval-def:EntityStateBoolType (0..1)	If the updatepersonalwebparts is enabled the perm

< spgroup_test >

The spgroup test is used to check the group properties for site collections. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an spwebapplication_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 838: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..1)	

< spgroup_object >

The spgroup_object element is used by a spgroup test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An spgroup object consists of a sitecollectionurl used to define a specific site collection. See the defintion of the SPGroup class in the SharePoint object model documentation.

Extends: oval-def:ObjectType

Child Elements

Table 839: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
sitecollectionurl	oval-def:EntityObjectStringType (1..1)	The sitecollectionurl element defines the Site Colection to evaluate specific group settings.
oval-def:filter	n/a (0..unbounded)	

< spgroup_state >

The spgroup_state element defines settings for groups in a site collections.

Extends: oval-def:StateType

Child Elements

Table 840: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
sitecollectionurl	oval-def:EntityStateStringType (0..1)	The sitecollectionurl element identifies a Site Collection.
gname	oval-def:EntityStateStringType (0..1)	The name element identifies a Group name.
autoacceptrequesttojoinleave	oval-def:EntityStateBoolType (0..1)	If the autoacceptrequesttojoinleave is enabled it allows users to automatically join groups.
allowmembersedit-membership	oval-def:EntityStateBoolType (0..1)	If the allowmemberseditmembership is enabled than all group memebers will be allowed to edit the membership of a group..
onlyallowmembersviewmembership	oval-def:EntityStateBoolType (0..1)	If the onlyallowmembersviewmembership is enabled it allows users to automatically join groups.

< spweb_test >

The spweb test is used to check the properties for site collections. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an spwebapplication_object and the optional state element specifies the data to check. See https://msdn.microsoft.com/en-us/library/ms473633.aspx for more information.

Extends: oval-def:TestType

Child Elements

Table 841: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..1)	

< spweb_object >

The spweb_object element is used by a spweb test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An spweb object consists of a webcollection url and sitecollection url used to define a specific web apoplication and a specific site collection. See the defintion of the SPWeb class in the SharePoint object model documentation.

Extends: oval-def:ObjectType

Child Elements

Table 842: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
webcollectionurl	oval-def:EntityObjectStringType (1..1)	Specifies a web site (this is the SPWeb object we want).
sitecollectionurl	oval-def:EntityObjectStringType (1..1)	Specifies a site collection.
oval-def:filter	n/a (0..unbounded)	

< spweb_state >

The spweb_state element defines settings for a site collection.

Extends: oval-def:StateType

Child Elements

Table 843: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
webcollectionurl	oval-def:EntityStateStringType (0..1)	The webcollectionurl specifies a web site (the SPWeb object).
sitecollectionurl	oval-def:EntityStateStringType (0..1)	The sitecollectionurl element specifies a site collection.
secondarysitecolladmin	oval-def:EntityStateStringType (0..1)	The secondarysitecolladmin element identifies a secondary site collection admin.
secondsitecolladminenabled	oval-def:EntityStateBoolType (0..1)	A boolean that represents if the secondarysitecolladmin is enabled.
allowanonymousaccess	oval-def:EntityStateBoolType (0..1)	If the allowanonymousaccess is enabled users will be allowed to create and manager their own top-level Web sites .

< splist_test >

The splist test is used to check the properties of lists associated with a SharePoint site or site collection. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an splist_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 844: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..1)	

< splist_object >

The splist_object element is used by a splist test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An splist object consists of a spsiteurl used to define a specific site in a site collection that various security related configuration items need to be checked. See the defintion of the SPList class in the SharePoint object model documentation.

Extends: oval-def:ObjectType

Child Elements

Table 845: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
spsiteurl	oval-def:EntityObjectStringType (1..1)	The spsiteurl element defines the Sharepoint website being specified …
oval-def:filter	n/a (0..unbounded)	

< splist_state >

The splist_state element defines the different information that can be used to evaluate the specified Sharepoint sites. . . .

Extends: oval-def:StateType

Child Elements

Table 846: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
spsiteurl	oval-def:EntityStateStringType (0..1)	The spsiteurl element identifies an Sharepoint site to test for.
irmenabled	oval-def:EntityStateBoolType (0..1)	If the irmenabled option is enabled, documents are protected whenever they leave the control of the Sharepoint system.
enableversioning	oval-def:EntityStateBoolType (0..1)	If the enableversioning option is enabled, backup copies of documents are kept and managed by the Sharepoint system.
nocrawl	oval-def:EntityStateBoolType (0..1)	If the nocrawl option is enabled, the site is excluded from crawls that Sharepoint does when it indexes sites.

< spantivirussettings_test >

The spantivirussettings test is used to check the settings for antivirus software associated with a SharePoint deployment.

Extends: oval-def:TestType

Child Elements

Table 847: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..1)	

< spantivirussettings_object >

The spantivirussettings_object element is used by a spantivirussettings test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An spantivirussettings object consists of a spwebservicename used to define a specific webservice in a farm that various security related configuration items need to be checked and an spfarmname which denotes the farm of which the spwebservice is a part. See the defintion of the SPAntiVirusSettings class in the SharePoint object model documentation.

Extends: oval-def:ObjectType

Child Elements

Table 848: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
spwebservicename	oval-def:EntityObjectStringType (1..1)	The spwebservicename element denotes the web service for which antivirus settings will be checked.
spfarmname	oval-def:EntityObjectStringType (1..1)	The spfarmname element denotes the farm on which a web service to be queried resides.
oval-def:filter	n/a (0..unbounded)	

< spantivirussettings_state >

The spantivirus_state element defines the different information that can be used to evaluate the specified Sharepoint sites. . . .

Extends: oval-def:StateType

Child Elements

Table 849: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
spwebservicename	oval-def:EntityStateStringType (0..1)	The spwebservicename denotes the name of a SharePoint web service to be tested or * (the default) to test all web services.
spfarmname	oval-def:EntityStateStringType (0..1)	The spfarmname denotes the name of the farm on which the Sharepoint webservice resides or the local farm (default).
allowdownload	oval-def:EntityStateBoolType (0..1)	Specifies whether infected documents can be downloaded on the SharePoint system.
cleaningenabled	oval-def:EntityStateBoolType (0..1)	Specifies whether the virus scanner should attempt to cure files that are infected.
downloadscanenabled	oval-def:EntityStateBoolType (0..1)	Specifies whetehr files are scanned for viruses when they are downloaded.
numberofthreads	oval-def:EntityStateIntType (0..1)	The number of threads that the antivirus scanner can use to scan documents for viruses.
skipsearchcrawl	oval-def:EntityStateBoolType (0..1)	Specifies whether to skip scanning for viruses during a search crawl.
timeout	oval-def:EntityStateIntType (0..1)	Denotes the amount of time before the virus scanner times out in seconds.
uploadscanenabled	oval-def:EntityStateBoolType (0..1)	Specifies whether files are scanned when they are uploaded.
vendorupdatecount	oval-def:EntityStateIntType (0..1)	Denotes the current increment of the number of times the vendor has been updated.

< spsiteadministration_test >

The spsiteadministration test is used to check the properties of a site. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an spwebapplication_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 850: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..1)	

< spsiteadministration_object >

The spsiteadministration_object element is used by a spsiteadministration test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An spsiteadministration object consists of a webapplicationurl used to define a specific web application. The collected data is available via the SPQuota class, which can be found via the SPSite object. See the defintions of the SPSite and the SPQuota classes in the SharePoint object model documentation.

Extends: oval-def:ObjectType

Child Elements

Table 851: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
sitecollectionurl	oval-def:EntityObjectStringType (1..1)	The sitecollectionurl element defines the site to evaluate.
oval-def:filter	n/a (0..unbounded)	

< spsiteadministration_state >

The spspsiteadministration_state element defines security settings and permissions that can be checked for a specified SPSite.

Extends: oval-def:StateType

Child Elements

Table 852: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
sitecollectionurl	oval-def:EntityStateStringType (0..1)	The sitecollectionurl element identifies a site.
storagemaxlevel	oval-def:EntityStateIntType (0..1)	The storagemaxlevel is the maximum storage allowed for the site.
storagewarninglevel	oval-def:EntityStateIntType (0..1)	When the storagewarninglevel is reached a site collection receive advance notice before available storage is expended.s.

< spsite_test >

The spsite test is used to check the properties of a site. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an spwebapplication_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 853: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..1)	

< spsite_object >

The spsite_object element is used by a spsiteadministration test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An spsite object consists of a sitecollectionurl used to define a specific web application. See the defintion of the SPSite class in the SharePoint object model documentation.

Extends: oval-def:ObjectType

Child Elements

Table 854: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
sitecollectionurl	oval-def:EntityObjectStringType (1..1)	The sitecollectionurl element defines the site to evaluate.
oval-def:filter	n/a (0..unbounded)	

< spsite_state >

The spsite_state element defines security settings and permissions that can be checked for a specified SPSite.

Extends: oval-def:StateType

Child Elements

Table 855: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
sitecollectionurl	oval-def:EntityStateStringType (0..1)	The sitecollectionurl element identifies a site.
quotaname	oval-def:EntityStateStringType (0..1)	The quota name is the name of quota template for a site collection.
url (Deprecated)	oval-def:EntityStateStringType (0..1)	The URL is the full URL to the root Web site of the site collection, including host name, port number, and path.

< spcrawlrule_test >

The spcrawlrule test is used to check the configuration or rules associated with the SharePoint system's built-in indexer and the sites or documents that will be indexed.

Extends: oval-def:TestType

Child Elements

Table 856: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..1)	

< spcrawlrule_object >

The spcrawlrule_object element is used by a spcrawlrule test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An spcrawlrule object consists of a spsiteurl used to define a specific resource (eg. website or document) on a server that can be indexed by the SharePoint indexer. See the defintion of the CrawlRule class in the SharePoint object model documentation.

Extends: oval-def:ObjectType

Child Elements

Table 857: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
spsiteurl	oval-def:EntityObjectStringType (1..1)	The spsiteurl element denotes the resource on the SharePoint server (eg. a site or document) for which indexing settings will be checked.
oval-def:filter	n/a (0..unbounded)	

< spcrawlrule_state >

The spcrawlrule state element defines the various properties of the SharePoint indexer that can be checked.

Extends: oval-def:StateType

Child Elements

Table 858: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
spsiteurl	oval-def:EntityStateStringType (0..1)	The spsiteurl denotes the URL of a website or resource whose indexing properties should be tested.
crawlashttp	oval-def:EntityStateBoolType (0..1)	Specifies whether the crawler should crawl content from a hierarchical content source, such as HTTP content.
enabled	oval-def:EntityStateBoolType (0..1)	Specifies whether a particular crawl rule is enabled.
followcomplexurls	oval-def:EntityStateBoolType (0..1)	Specifies whether the indexer should crawl websites that contain the question mark (?) character.
path	oval-def:EntityStateStringType (0..1)	The path to which a particular crawl rule applies.
priority	oval-def:EntityStateIntType (0..1)	The priority setting for a particular crawl rule.
suppressindexing	oval-def:EntityStateBoolType (0..1)	Specifies whether the crawler should exclude the content of items that this rule applies to from the content index.
accountname	oval-def:EntityStateStringType (0..1)	A string containing the account name for the crawl rule.

< spjobdefinition_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.10
- Reason: Replaced by the spjobdefinition510_test. This test does not uniquely identify a single job definition. A new test was created to use displaynames, which are unique. See the spjobdefinition510_test.
- Comment: This test has been deprecated and will be removed in version 6.0 of the language.

The spjobdefinition test is used to check the status of the various properties associated with scheduled jobs in the SharePoint system.

Extends: oval-def:TestType

Child Elements

Table 859: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..1)	

< spjobdefinition_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.10
- Reason: Replaced by the spjobdefinition510_object. This test does not uniquely identify a single job definition. A new object was created to use displaynames, which are unique. See the spjobdefinition510_object.
- Comment: This test has been deprecated and will be removed in version 6.0 of the language.

The spjobdefinition_object element is used by a spjobdefinition test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An spjobdefinition_object consists of a webappuri used to define a specific web application for which job checks should be done. See the defintion of the SPJobDefinition class in the SharePoint object model documentation.

Extends: oval-def:ObjectType

Child Elements

Table 860: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
webappuri	oval-def:EntityObjectStringType (1..1)	The URI that represents the web application for which jobs should be checked.
oval-def:filter	n/a (0..unbounded)	

< spjobdefinition_state > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.10
- Reason: Replaced by the spjobdefinition510_state. This state does not uniquely identify a single job definition. A new state was created to use displaynames, which are unique. See the spjobdefinition510_state.
- Comment: This test has been deprecated and will be removed in version 6.0 of the language.

The various properties of a Sharepoint job that can be checked.

Extends: oval-def:StateType

Child Elements

Table 861: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
webappuri	oval-def:EntityStateStringType (0..1)	The URI that represents the web application for which jobs should be checked.
displayname	oval-def:EntityStateStringType (0..1)	The name of the job as displayed in the SharePoint Central Administration site.
isdisabled	oval-def:EntityStateBoolType (0..1)	Determines whether or not the job definition is enabled.
retry	oval-def:EntityStateBoolType (0..1)	Determines whether the job definition should be retried if it ends abnormally.
title	oval-def:EntityStateStringType (0..1)	The title of a job as displayed in the SharePoint Central Administration site.

< spjobdefinition510_test >

The spjobdefinition test is used to check the status of the various properties associated with scheduled jobs in the SharePoint system.

Extends: oval-def:TestType

Child Elements

Table 862: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..1)	

< spjobdefinition510_object >

The spjobdefinition510_object element is used by a spjobdefinition test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An spjobdefinition510_object consists of a webappuri and displayname used to define a specific web application for which job checks should be done. See the defintion of the SPJobDefinition class in the SharePoint object model documentation.

Extends: oval-def:ObjectType

Child Elements

Table 863: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
webappuri	oval-def:EntityObjectStringType (1..1)	The URI that represents the web application for which jobs should be checked.
displayname	oval-def:EntityObjectStringType (1..1)	The name of the job as displayed in the SharePoint Central Administration site.
oval-def:filter	n/a (0..unbounded)	

< spjobdefinition510_state >

The various properties of a Sharepoint job that can be checked.

Extends: oval-def:StateType

Child Elements

Table 864: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
webappuri	oval-def:EntityStateStringType (0..1)	The URI that represents the web application for which jobs should be checked.
displayname	oval-def:EntityStateStringType (0..1)	The name of the job as displayed in the SharePoint Central Administration site.
isdisabled	oval-def:EntityStateBoolType (0..1)	Determines whether or not the job definition is enabled.
retry	oval-def:EntityStateBoolType (0..1)	Determines whether the job definition should be retried if it ends abnormally.
title	oval-def:EntityStateStringType (0..1)	The title of a job as displayed in the SharePoint Central Administration site.

< bestbet_test >

The bestbet test is used to get all the best bets associated with a site.

Extends: oval-def:TestType

Child Elements

Table 865: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..1)	

< bestbet_object >

The bestbet_object element is used by a bestbet test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An bestbet object consists of a sitecollectionurl used to define a specific site and a bestbeturl used to define a specific best bet. See the defintion of the BestBet class in the SharePoint object model documentation.

Extends: oval-def:ObjectType

Child Elements

Table 866: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
sitecollectionurl	oval-def:EntityObjectStringType (1..1)	The URL that represents the site collection.
bestbeturl	oval-def:EntityObjectStringType (1..1)	The URL that represents the best bet.
oval-def:filter	n/a (0..unbounded)	

< bestbet_state >

The various properties of a Best Bet that can be checked.

Extends: oval-def:StateType

Child Elements

Table 867: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
sitecollectionurl	oval-def:EntityStateStringType (0..1)	The URL that represents the site collection.
bestbeturl	oval-def:EntityStateStringType (0..1)	The name of the job as displayed in the SharePoint Central Administration site.
title	oval-def:EntityStateStringType (0..1)	The title of a best bet.
description	oval-def:EntityStateStringType (0..1)	Thedescription of a best bet..

< infopolicycoll_test >

The policycoll test is used to get all the Information Policies associated with a site.

Extends: oval-def:TestType

Child Elements

Table 868: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..1)	

< infopolicycoll_object >

The infopolicycoll_object element is used by a policycoll test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A infopolicycoll object consists of a sitecollectionurl used to define a specific site and an id used to define a specific information policy. See the defintion of the Policy class and policycollection class in the SharePoint object model documentation.

Extends: oval-def:ObjectType

Child Elements

Table 869: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
sitecollectionurl	oval-def:EntityObjectStringType (1..1)	The URL that represents the site collection.
id	oval-def:EntityObjectStringType (1..1)	The id that represents the Information Policy.
oval-def:filter	n/a (0..unbounded)	

< infopolicycoll_state >

The various properties of the Information Policy that can be checked.

Extends: oval-def:StateType

Child Elements

Table 870: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
sitecollectionurl	oval-def:EntityStateStringType (0..1)	The URL that represents the site collection.
id	oval-def:EntityStateStringType (0..1)	The id of the Information Policy.
name	oval-def:EntityStateStringType (0..1)	The name of the Information Policy.
description	oval-def:EntityStateStringType (0..1)	The description of an Information Policy..
longdescription	oval-def:EntityStateStringType (0..1)	The long description of an Information Policy..

< spdiagnosticsservice_test >

The spdiagnosticsservice test is used to check the diagnostic properties associated with a Sharepoint system.

Extends: oval-def:TestType

Child Elements

Table 871: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..1)	

< spdiagnosticsservice_object >

The spdiagnosticsservice_object element is used by an spdiagnosticsservice test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An spdiagnosticsservice object consists of a farmname used to define a specific Sharepoint farm for which diagnostics properties should be checked. See the defintion of the SPDiagnosticsService class in the SharePoint object model documentation.

Extends: oval-def:ObjectType

Child Elements

Table 872: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
farmname	oval-def:EntityObjectStringType (1..1)	The farm whose diagnostic capabilities should be checked. Use .* for all farms or SPFarm.Local for the local farm.
oval-def:filter	n/a (0..unbounded)	

< spdiagnosticsservice_state >

The various properties of a diagnostics service that can be checked.

Extends: oval-def:StateType

Child Elements

Table 873: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
farmname	oval-def:EntityStateStringType (0..1)	The farm whose diagnostic capabilities should be checked.
displayname	oval-def:EntityStateStringType (0..1)	The name of the diagnostic service as shown in the Sharepoint Central Administration site.
logcutinterval	oval-def:EntityStateIntType (0..1)	The number of minutes to capture events to a single log file. This value lies in the range 0 to 1440. The default value is 30.
loglocation	oval-def:EntityStateStringType (0..1)	The path to the file system directory where log files are created and stored.
logstokeep	oval-def:EntityStateIntType (0..1)	The value that indicates the number of log files to create. This lies in the range 0 to 1024 with a default of 96.
required	oval-def:EntityStateBoolType (0..1)	The required property specifies whether an instance of the spdiagnosticsservice must be running on the farm.
typename	oval-def:EntityStateStringType (0..1)	The friendly name for the service as displayed in the Central Administration and in logs. This should be “Windows Sharepoint Diagnostics Service” by default.

< spdiagnosticslevel_test >

The spdiagnosticslevel_test is used to check the status of the logging features associated with a Sharepoint deployment.

Extends: oval-def:TestType

Child Elements

Table 874: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..1)	

< spdiagnosticslevel_object >

The spdiagnosticslevel_object element is used by an spdiagnosticslevel test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An spdiagnosticslevel object consists of a farmname used to define a specific Sharepoint farm for which policy properties should be checked. See the defintion of the SPWebApplication class in the SharePoint object model documentation. See the defintion of the IDiagnosticsLevel Interface in the SharePoint object model documentation.

Extends: oval-def:ObjectType

Child Elements

Table 875: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
farmname	oval-def:EntityObjectStringType (1..1)	The farm whose diagnostics levels should be checked. Use .* for all farms or SPFarm.Local for the local farm.
oval-def:filter	n/a (0..unbounded)	

< spdiagnosticslevel_state >

The various properties of a Diagnostics level that can be checked.

Extends: oval-def:StateType

Child Elements

Table 876: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
farmname	oval-def:EntityStateStringType (0..1)	The name of the farm for which diagnostics level properties should be checked.
event-severity	sp-def:EntityStateEventSeverityType (0..1)	The event severity setting for a particular diagnostic level category.
hidden	oval-def:EntityStateBoolType (0..1)	Specifies whether the trace log category is hidden in the Windows Sharepoint Services Central Administration interface.
levelid	oval-def:EntityStateStringType (0..1)	A string that represents the ID of the trace log category. This is its English language name.
level-name	oval-def:EntityStateStringType (0..1)	The name of the trace log category. This represents the localized name for the category.
trace-severity	sp-def:EntityStateTraceSeverityType (0..1)	The trace severity setting for a particular diagnostic level category.

< sppolicyfeature_test >

The sppolicyfeature test enables one to check the attributes associated with policies and policy features on the Sharepoint deployment.

Extends: oval-def:TestType

Child Elements

Table 877: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..1)	

< sppolicyfeature_object >

The sppolicyfeature_object element is used by an sppolicyfeature test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An sppolicyfeature object consists of a farmname used to define a specific Sharepoint farm for which policy feature properties should be checked. See the defintion of the PolicyFeature class in the SharePoint object model documentation.

Extends: oval-def:ObjectType

Child Elements

Table 878: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
farmname	oval-def:EntityObjectStringType (1..1)	The farm whose policy features should be checked. Use .* for all farms or SPFarm.Local for the local farm.
oval-def:filter	n/a (0..unbounded)	

< sppolicyfeature_state >

The various properties of a policy feature that can be checked.

Extends: oval-def:StateType

Child Elements

Table 879: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
farmname	oval-def:EntityStateStringType (0..1)	The farm whose policy features should be checked. Use .* for all farms or SPFarm.Local for the local farm.
config-page	oval-def:EntityStateStringType (0..1)	The URL to a web control used to edit policy instance-level settings.
default-custom-data	oval-def:EntityStateStringType (0..1)	The default values for any policy instance-level settings for a policy feature.
description	oval-def:EntityStateStringType (0..1)	The short description of the policy feature and of the service it provides.
global-config-page	oval-def:EntityStateStringType (0..1)	The URL to a web control used to edit server farm-level settings for this policy feature.
globalcustomdata	oval-def:EntityStateStringType (0..1)	The default settings for any server farm-level settings for this policy feature.
group	oval-def:EntityStateStringType (0..1)	The policy feature group to which a policy feature belongs.
name	oval-def:EntityStateStringType (0..1)	The name to display in the Microsoft Office Sharepoint Server 2007 interface for an information policy feature.
publisher	oval-def:EntityStateStringType (0..1)	The name of the creator of the policy feature as it is displayed in the Microsoft Office Sharepoint Server 2007 user interface.
state	sp-def:EntityStatePolicyFeatureStateType (0..1)	Specifies whether the policy feature is hidden or visible.

< sppolicy_test >

The sppolicy test enables one to check the attributes of the policies associated with a particular URL Zone in a Sharepoint system.

Extends: oval-def:TestType

Child Elements

Table 880: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..1)	

< sppolicy_object >

The sppolicy_object element is used by an sppolicy test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An sppolicy object consists of a webappuri and a URL Zone used to define a specific Sharepoint web application and zone for which policy properties should be checked. See the defintion of the SPPolicy class and the sppolicyroletype in the SharePoint object model documentation.

Extends: oval-def:ObjectType

Child Elements

Table 881: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
webappuri	oval-def:EntityObjectStringType (1..1)	The URI that represents the web application for which policies should be checked.
urlzone	sp-def:EntityObjectUrlZoneType (1..1)	The zone for which policies should be checked.

< sppolicy_state >

The various properties of a policy that can be checked.

Extends: oval-def:StateType

Child Elements

Table 882: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
webappuri	oval-def:EntityStateStringType (0..1)	The URI that represents the web application for which policies should be checked.
urlzone	sp-def:EntityStateUrlZoneType (0..1)	The zone for which policies should be checked.
displayname	oval-def:EntityStateStringType (0..1)	The user or group display name for a policy. This defaults to the user name if the display name cannot be resolved through Active Directory.
issystemuser	oval-def:EntityStateBoolType (0..1)	Specifies whether the user identified by a particular policy is visible only as a System account within the Windows Sharepoint Services user interface.
username	oval-def:EntityStateStringType (0..1)	The user name of the user or group that is associated with policy.
policyroletype	sp-def:EntityStatePolicyRoleType (0..1)	The policy role type to apply globally in a Sharepoint web application to a user or group.

== EntityObjectUrlZoneType ==

The EntityObjectUrlZoneType restricts a string value to a set of values that describe the different IIS Url Zones. The empty string is also allowed to support empty element associated with error conditions.

Restricts: oval-def:EntityObjectStringType

Table 883: Enumeration Values

Value	Description
Custom	(No Description)
Default	(No Description)
Extranet	(No Description)
Intranet	(No Description)
Internet	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateEventSeverityType ==

The EntityStateEventSeverityType restricts a string value to a set of values that describe the different states that can be configured for a diagnostics level event severity level property of the diagnostics service.

Restricts: oval-def:EntityStateStringType

Table 884: Enumeration Values

Value	Description
Error	(No Description)
ErrorCritical	(No Description)
ErrorSecurityBreach	(No Description)
ErrorServiceUnavailable	(No Description)
FailureAudit	(No Description)
Information	(No Description)
None	(No Description)
Success	(No Description)
SuccessAudit	(No Description)
Warning	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateTraceSeverityType ==

The EntityStateTraceSeverityType restricts a string value to a set of values that describe the different states that can be configured for a diagnostics level trace severity level property of the diagnostics service.

Restricts: oval-def:EntityStateStringType

Table 885: Enumeration Values

Value	Description
High	(No Description)
Medium	(No Description)
Monitorable	(No Description)
None	(No Description)
Unexpected	(No Description)
Verbose	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStatePolicyRoleType ==

The EntityStatePolicyRoleType restricts a string value to a set of values that describe the different Policy settings for Access Control that are available for users.

Restricts: oval-def:EntityStateStringType

Table 886: Enumeration Values

Value	Description
DenyAll	Deny all rights.
DenyWrite	Deny write permissions.
FullControl	Grant full control.
FullRead	Grant full read permissions.
None	No role type assigned.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStatePolicyFeatureStateType ==

The EntityStatePolicyRoleType restricts a string value to a set of values that describe the different policy feature states that can be configured for a policy feature.

Restricts: oval-def:EntityStateStringType

Table 887: Enumeration Values

Value	Description
Hidden	Specifies that the policy feature is hidden from the Sharepoint Central Administration user interface.
Visible	Specifies that the policy feature is visible from the Sharepoint Central Administration user interface.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateUrlZoneType ==

The EntityStateUrlZoneType restricts a string value to a set of values that describe the different IIS Url Zones.

Restricts: oval-def:EntityStateStringType

Table 888: Enumeration Values

Value	Description
Custom	(No Description)
Default	(No Description)
Extranet	(No Description)
Intranet	(No Description)
Internet	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

Open Vulnerability and Assessment Language: SharePoint System Characteristics

- Schema: SharePoint System Characteristics
- Version: 5.11.1:1.1
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the SharePoint specific system characteristic items found in Open Vulnerability and Assessment Language (OVAL). Each item is an extension of the standard item element defined in the Core System Characteristic Schema. Through extension, each item inherits a set of elements and attributes that are shared amongst all OVAL Items. Each item is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core System Characteristic Schema is not outlined here.

The SharePoint Component Schema is based on the SharePoint Object Model (Windows SharePoint Services 3.0)

The OVAL Schema is maintained by the 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.cisecurity.org.

Item Listing

- *< spwebapplication_item >*
- *< spgroup_item >*
- *< spweb_item >*
- *< splist_item >*
- *< spantivirussettings_item >*
- *< spsiteadministration_item >*
- *< spsite_item >*
- *< spcrawlrule_item >*

- *< spjobdefinition_item > (Deprecated)*
- *< spjobdefinition510_item >*
- *< bestbet_item >*
- *< infopolicycoll_item >*
- *< spdiagnosticsservice_item >*
- *< spdiagnosticslevel_item >*
- *< sppolicyfeature_item >*
- *< sppolicy_item >*

< spwebapplication_item >

This spwebapplication item stores information for security related features and permissions related to each web application. See the defintion of the SPWebApplication class in the SharePoint object model documentation.

Extends: oval-sc:ItemType

Child Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
webapplicationurl	oval-sc:EntityItemStringType (0..1)	A string the represents the url that identif
allowparttopartcommunication	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if a user can cre
allowaccesstowebpartcatalog	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if a user can cre
blockedfileextention	oval-sc:EntityItemStringType (0..unbounded)	A single blockedfileextention for the appl
defaultquotatemplate	oval-sc:EntityItemStringType (0..1)	A string the represents the default quota t
externalworkflowparticipantsenabled	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if a user is allo
recyclebinenabled	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the recycle bi
automaticallydeleteunusedsitecollections	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the site can b
selfservicesitecreationenabled	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if a self service
secondstagerecyclebinquota	oval-sc:EntityItemIntType (0..1)	Size of the second stage recycle bin quota
recyclebinretentionperiod	oval-sc:EntityItemIntType (0..1)	The recyclebinretentionperiod is the reten
outboundmailserverinstance	oval-sc:EntityItemStringType (0..1)	The string name of the outboundmailserv
outboundmailsenderaddress	oval-sc:EntityItemStringType (0..1)	The from address that is used when sendi
outboundmailreplytoaddress	oval-sc:EntityItemStringType (0..1)	The reply to address that is used when se
secvalexpires	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if a security val
timeout	oval-sc:EntityItemIntType (0..1)	The timeout is the amount of time before
isadministrationwebapplication	oval-sc:EntityItemBoolType (0..1)	A boolean that specifies whether the curr
applicationpoolname	oval-sc:EntityItemStringType (0..1)	A string that represents the application po
applicationpoolusername	oval-sc:EntityItemStringType (0..1)	A string that represents the application po
openitems	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio
addlistitems	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio
approveitems	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio
deletelistitems	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio
deleteversions	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio

Ta

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
editlistitems	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if edit items in
managelists	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio
viewversions	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio
viewlistitems	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio
cancelcheckout	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio
createalerts	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio
viewformpages	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio
viewpages	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio
addandcustomizepages	oval-sc:EntityItemBoolType (0..1)	
applystylesheets	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio
applythemeandborder	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio
browsedirectories	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio
browseuserinfo	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio
creategroups	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio
createsscsite	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio
editmyuserinfo	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio
enumeratepermissions	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio
managealerts	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio
managepermissions	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio
managesubwebs	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio
manageweb	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio
open	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio
useclientintegration	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio
useremoteapis	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio
viewusagedata	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio
managepersonalviews	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio
adddelprivatewebparts	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio
updatepersonalwebparts	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the permissio

< spgroup_item >

This spgroup item stores information for security related features related to site groups

Extends: oval-sc:ItemType

Child Elements

Table 890: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
sitecollectionurl	oval-sc:EntityItemStringType (0..1)	A string the represents the url that identifies the site collection.
gname	oval-sc:EntityItemStringType (0..1)	A string the represents the name of a group in a site collection.
autoacceptrequesttojoinleave	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if sites can automatically accepts requests.
allowmembersedit-membership	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if owners other than the group owner can edit the membership of groups.
onlyallowmembersviewmembership	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if owners other than the group owner can edit the membership of groups.

< spweb_item >

This spweb item stores information for security related features related to site collections.

Extends: oval-sc:ItemType

Child Elements

Table 891: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
webcollectionurl	oval-sc:EntityItemStringType (0..1)	A string that specifies a web site (the SPWeb object).
sitecollectionurl	oval-sc:EntityItemStringType (0..1)	A string that specifies a site collection.
secondarysitecolladmin	oval-sc:EntityItemStringType (0..1)	A string the represents the secondarysitecolladmin.
secondsitecolladminenabled	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if the secondsitecolladmin is enabled.
allowanonymousaccess	oval-sc:EntityItemBoolType (0..1)	A boolean that represents if a anonymous access is allowed to the web site.

< splist_item >

An SPList represents a list of content on a Sharepoint web site. It consists of items or rows and columns or fields that contain data.

Extends: oval-sc:ItemType

Child Elements

Table 892: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
spsiteurl	oval-sc:EntityItemStringType (0..1)	The url that identifies the website.
irmenabled	oval-sc:EntityItemBoolType (0..1)	The irmenabled attribute tests to see if documents that leave the Sharepoint environment are protected.
enableversioning	oval-sc:EntityItemBoolType (0..1)	The enableversioning attribute specifies whether backup copies of files should be created and managed in the Sharepoint system.
nocrawl	oval-sc:EntityItemBoolType (0..1)	The nocrawl attribute indicates that this site should not be among those crawled and indexed.

< spantivirussettings_item >

An SPAntivirusSettings Item represents the set of antivirus-related security settings on a Sharepoint server.

Extends: oval-sc:ItemType

Child Elements

Table 893: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
spwebservicename	oval-sc:EntityItemStringType (0..1)	The name of the SP Web Service for which to retrieve the antivirus settings or * for all web services. The default value is * which checks all SP Web services
spfarmname	oval-sc:EntityItemStringType (0..1)	The Farm in which the SP Web Service resides.
allowdownload	oval-sc:EntityItemBoolType (0..1)	Specifies whether SharePoint users can download documents that are found to be infected.
cleaningenabled	oval-sc:EntityItemBoolType (0..1)	Specifies whether or not the virus scanner should attempt to cure infected files.
downloadscanenabled	oval-sc:EntityItemBoolType (0..1)	Specifies whether files are scanned when they are downloaded.
numberofthreads	oval-sc:EntityItemIntType (0..1)	Specifies the number of threads that the virus scanner may use to perform virus scans.
skipsearchcrawl	oval-sc:EntityItemBoolType (0..1)	Specifies whether to skip document virus scanning during a search crawl.
timeout	oval-sc:EntityItemIntType (0..1)	The amount of time before the virus scanner times out in seconds.
uploadscanenabled	oval-sc:EntityItemBoolType (0..1)	Specifies whether files are scanned for viruses when they are uploaded.
vendorupdatecount	oval-sc:EntityItemIntType (0..1)	The current increment of the number of times the vendor has been updated.

< spsiteadministration_item >

This spsiteadministration item stores information for security related features and permissions related to each top-level web sites. See the defintion of the SPSiteAdministration class in the SharePoint object model documentation.

Extends: oval-sc:ItemType

Child Elements

Table 894: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
sitecollectionurl	oval-sc:EntityItemStringType (0..1)	A string the represents the url that identifies the sitecollection application.
storagemaxlevel	oval-sc:EntityItemIntType (0..1)	The storagemaxlevel is the maximum storage allowed for the site.
storagewarninglevel	oval-sc:EntityItemIntType (0..1)	When the storagewarninglevel is reached a site collection receive advance notice before available storage is expended.

< spsite_item >

This spsite item stores information for security related features for sites. See the defintion of the SPSite class in the SharePoint object model documentation.

Extends: oval-sc:ItemType

Child Elements

Table 895: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
sitecollectionurl	oval-sc:EntityItemStringType (0..1)	A string the represents the url that identifies the sitecollection application.
quotaname	oval-sc:EntityItemStringType (0..1)	The string that represents the name of the quota for a specific site collection.
url	oval-sc:EntityItemStringType (0..1)	

< spcrawlrule_item >

The spcrawlrule_item specifies rules that the SharePoint system follows when it crawls the content of sites stored within it.

Extends: oval-sc:ItemType

Child Elements

Table 896: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
spsiteurl	oval-sc:EntityItemStringType (0..1)	A URL that represents the resource (eg. sites, documents,etc.) on which the crawl-rule tests should be run or * if the check should be run on all sites/documents on the server.
crawlashttp	oval-sc:EntityItemBoolType (0..1)	Specifies whether the crawler should crawl content from a hierarchical content source, such as HTTP content.
enabled	oval-sc:EntityItemBoolType (0..1)	Specifies whether a particular crawl rule is enabled.
followcomplexurls	oval-sc:EntityItemBoolType (0..1)	Specifies whether the indexer should crawl websites that contain the question mark (?) character.
path	oval-sc:EntityItemStringType (0..1)	The path to which a particular crawl rule applies.
priority	oval-sc:EntityItemIntType (0..1)	The priority setting for a particular crawl rule.
suppressindexing	oval-sc:EntityItemBoolType (0..1)	Specifies whether the crawler should exclude the content of items that this rule applies to from the content index.
accountname	oval-sc:EntityItemStringType (0..1)	A string containing the account name for the crawl rule.

< spjobdefinition_item > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.10
- Reason: Replaced by the spjobdefinition510_item. This item does not uniquely identify a single job definition. A new state was created to use displaynames, which are unique. See the spjobdefinition510_item.
- Comment: This item has been deprecated and may be removed in a future version of the language.

This represents the set of Job Definitions that are scheduled to run on each SharePoint Web Application

Extends: oval-sc:ItemType

Child Elements

Table 897: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
webappuri	oval-sc:EntityItemStringType (0..1)	The URI that represents the web application for which the IIS Settings should be checked.
displayname	oval-sc:EntityItemStringType (0..1)	The name of the job as displayed in the SharePoint Central Administration site.
isdisabled	oval-sc:EntityItemBoolType (0..1)	Determines whether or not the job definition is enabled.
retry	oval-sc:EntityItemBoolType (0..1)	Determines whether the job definition should be retried if it ends abnormally.
title	oval-sc:EntityItemStringType (0..1)	The title of a job as displayed in the SharePoint Central Administration site.

< spjobdefinition510_item >

This represents the set of Job Definitions that are scheduled to run on each SharePoint Web Application

Extends: oval-sc:ItemType

Child Elements

Table 898: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
webappuri	oval-sc:EntityItemStringType (0..1)	The URI that represents the web application for which the IIS Settings should be checked.
displayname	oval-sc:EntityItemStringType (0..1)	The name of the job as displayed in the SharePoint Central Administration site.
isdisabled	oval-sc:EntityItemBoolType (0..1)	Determines whether or not the job definition is enabled.
retry	oval-sc:EntityItemBoolType (0..1)	Determines whether the job definition should be retried if it ends abnormally.
title	oval-sc:EntityItemStringType (0..1)	The title of a job as displayed in the SharePoint Central Administration site.

< bestbet_item >

This represents the set of Best Bets for a site collection.

Extends: oval-sc:ItemType

Child Elements

Table 899: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
sitecollectionurl	oval-sc:EntityItemStringType (0..1)	The sitecollectionurl represents the URL for the site.
bestbeturl	oval-sc:EntityItemStringType (0..1)	The bestbeturl represents the URL for the best bet.
title	oval-sc:EntityItemStringType (0..1)	The title of the Best Bet.
description	oval-sc:EntityItemStringType (0..1)	The description of the Best Bet.

< infopolicycoll_item >

This represents the set of Information Policies for a site collection.

Extends: oval-sc:ItemType

Child Elements

Table 900: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
sitecollectionurl	oval-sc:EntityItemStringType (0..1)	The sitecollectionurl represents the URL for the site.
id	oval-sc:EntityItemStringType (0..1)	The id of the sitecollection poilicy.
name	oval-sc:EntityItemStringType (0..1)	The name of the sitecollection poilicy.
description	oval-sc:EntityItemStringType (0..1)	The description of the Information Policy.
longdescription	oval-sc:EntityItemStringType (0..1)	The long description of an Information Policy.

< spdiagnosticsservice_item >

This represents the set of diagnostic capabilities for Windows Sharepoint Services.

Extends: oval-sc:ItemType

Child Elements

Table 901: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
farm-name	oval-sc:EntityItemStringType (0..1)	The farm whose diagnostic capabilities should be checked. Use .* for all farms or SPFarm.Local for the local farm.
display-name	oval-sc:EntityItemStringType (0..1)	The name of the diagnostic service as shown in the Sharepoint Central Administration site.
log-cutinterval	oval-sc:EntityItemIntType (0..1)	The number of minutes to capture events to a single log file. This value lies in the range 0 to 1440. The default value is 30.
loglocation	oval-sc:EntityItemStringType (0..1)	The path to the file system directory where log files are created and stored.
logsto-keep	oval-sc:EntityItemIntType (0..1)	The value that indicates the number of log files to create. This lies in the range 0 to 1024 with a default of 96.
required	oval-sc:EntityItemBoolType (0..1)	The required property specifies whether an instance of the spdiagnosticsservice must be running on the farm.
type-name	oval-sc:EntityItemStringType (0..1)	The friendly name for the service as displayed in the Central Administration and in logs. This should be “Windows Sharepoint Diagnostics Service” by default.

< spdiagnosticslevel_item >

The diagnostics level associated with a particular instance of a diagnostics service on a Sharepoint farm.

Extends: oval-sc:ItemType

Child Elements

Table 902: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
farmname	oval-sc:EntityItemStringType (0..1)	The farm whose diagnostics levels should be checked. Use .* for all farms or SPFarm.Local for the local farm.
event-severity	sp-sc:EntityItemEventSeverityType (0..1)	The event severity setting for a particular diagnostic level category.
hidden	oval-sc:EntityItemBoolType (0..1)	Specifies whether the trace log category is hidden in the Windows Sharepoint Services Central Administration interface.
levelid	oval-sc:EntityItemStringType (0..1)	A string that represents the ID of the trace log category. This is its English language name.
levelname	oval-sc:EntityItemStringType (0..1)	The name of the trace log category. This represents the localized name for the category.
trace-severity	sp-sc:EntityItemTraceSeverityType (0..1)	The trace severity setting for a particular diagnostic level category.

< sppolicyfeature_item >

This represents a policy feature that is installed on the Sharepoint server farm.

Extends: oval-sc:ItemType

Child Elements

Table 903: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
farmname	oval-sc:EntityItemStringType (0..1)	The farm whose policy features should be checked. Use .* for all farms or SPFarm.Local for the local farm.
config-page	oval-sc:EntityItemStringType (0..1)	The URL to a web control used to edit policy instance-level settings.
default-custom-data	oval-sc:EntityItemStringType (0..1)	The default values for any policy instance-level settings for a policy feature.
description	oval-sc:EntityItemStringType (0..1)	The short description of the policy feature and of the service it provides.
global-config-page	oval-sc:EntityItemStringType (0..1)	The URL to a web control used to edit server farm-level settings for this policy feature.
globalcustomdata	oval-sc:EntityItemStringType (0..1)	The default settings for any server farm-level settings for this policy feature.
group	oval-sc:EntityItemStringType (0..1)	The policy feature group to which a policy feature belongs.
name	oval-sc:EntityItemStringType (0..1)	The name to display in the Microsoft Office Sharepoint Server 2007 interface for an information policy feature.
publisher	oval-sc:EntityItemStringType (0..1)	The name of the creator of the policy feature as it is displayed in the Microsoft Office Sharepoint Server 2007 user interface.
state	sp-sc:EntityItemPolicyFeatureStateType (0..1)	Specifies whether the policy feature is hidden or visible.

< sppolicy_item >

This represents a policy on the Sharepoint system.

Extends: oval-sc:ItemType

Child Elements

Table 904: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
webappuri	oval-sc:EntityItemStringType (0..1)	The URI that represents the web application for which policies should be checked.
urlzone	sp-sc:EntityItemUrlZoneType (0..1)	The zone for which policies should be checked.
displayname	oval-sc:EntityItemStringType (0..1)	The user or group display name for a policy. This defaults to the user name if the display name cannot be resolved through Active Directory.
issystemuser	oval-sc:EntityItemBoolType (0..1)	Specifies whether the user identified by a particular policy is visible only as a System account within the Windows Sharepoint Services user interface.
username	oval-sc:EntityItemStringType (0..1)	The user name of the user or group that is associated with policy.
policyroletype	sp-sc:EntityItemPolicyRoleType (0..1)	The policy role type to apply globally in a Sharepoint web application to a user or group.

== EntityItemUrlZoneType ==

The EntityItemUrlZoneType restricts a string value to a set of values that describe the different IIS Url Zones. The empty string is also allowed to support empty element associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 905: Enumeration Values

Value	Description
Custom	(No Description)
Default	(No Description)
Extranet	(No Description)
Intranet	(No Description)
Internet	(No Description)
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemEventSeverityType ==

The EntityItemEventSeverityType restricts a string value to a set of values that describe the different states that can be configured for a diagnostics level event severity level property of the diagnostics service.

Restricts: oval-sc:EntityItemStringType

Table 906: Enumeration Values

Value	Description
Error	(No Description)
ErrorCritical	(No Description)
ErrorSecurityBreach	(No Description)
ErrorServiceUnavailable	(No Description)
FailureAudit	(No Description)
Information	(No Description)
None	(No Description)
Success	(No Description)
SuccessAudit	(No Description)
Warning	(No Description)
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemTraceSeverityType ==

The EntityItemTraceSeverityType restricts a string value to a set of values that describe the different states that can be configured for a diagnostics level trace severity level property of the diagnostics service.

Restricts: oval-sc:EntityItemStringType

Table 907: Enumeration Values

Value	Description
High	(No Description)
Medium	(No Description)
Monitorable	(No Description)
None	(No Description)
Unexpected	(No Description)
Verbose	(No Description)
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemPolicyFeatureStateType ==

The EntityItemPolicyFeatureStateType restricts a string value to a set of values that describe the different states that can be configured for a policy feature.

Restricts: oval-sc:EntityItemStringType

Table 908: Enumeration Values

Value	Description
Hidden	Specifies that the policy feature is hidden from the Sharepoint Central Administration user interface.
Visible	Specifies that the policy feature is visible from the Sharepoint Central Administration user interface.
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemPolicyRoleType ==

The EntityItemPolicyRoleType restricts a string value to a set of values that describe the different Policy settings for Access Control that are available for users.

Restricts: oval-sc:EntityItemStringType

Table 909: Enumeration Values

Value	Description
DenyAll	Deny all rights.
DenyWrite	Deny write permissions.
FullControl	Grant full control.
FullRead	Grant full read permissions.
None	No role type assigned.
	The empty string value is permitted here to allow for detailed error reporting.

Open Vulnerability and Assessment Language: UNIX Definition

- Schema: UNIX Definition
- Version: 5.11.1:1.2
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose generic UNIX tests found in Open Vulnerability and Assessment Language (OVAL). Each test is an extension of the standard test element defined in the Core Definition Schema. Through extension, each test inherits a set of elements and attributes that are shared amongst all OVAL tests. Each test is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core Definition Schema is not outlined here.

The OVAL Schema is maintained by the 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.cisecurity.org.

Test Listing

- *< dnscache_test >*
- *< file_test >*
- *< fileextendedattribute_test >*
- *< gconf_test >*
- *< inetd_test >*
- *< interface_test >*
- *< password_test >*
- *< process_test > (Deprecated)* (Deprecated)
- *< process58_test >*
- *< routingtable_test >*
- *< runlevel_test >*
- *< sccs_test > (Deprecated)* (Deprecated)
- *< shadow_test >*
- *< symlink_test >*
- *< sysctl_test >*
- *< uname_test >*
- *< xinetd_test >*

< dnscache_test >

The dnscache_test is used to check the time to live and IP addresses associated with a domain name. The time to live and IP addresses for a particular domain name are retrieved from the DNS cache on the local system. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a dnscache_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 910: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< dnscache_object >

The dnscache_object is used by the dnscache_test to specify the domain name(s) that should be collected from the DNS cache on the local system. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 911: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
domain_name	oval-def:EntityObjectStringType (1..1)	The domain_name element specifies the domain name(s) that should be collected from the DNS cache on the local system.
oval-def:filter	n/a (0..unbounded)	

< dnscache_state >

The dnscache_state contains three entities that are used to check the domain name, time to live, and IP addresses associated with the DNS cache entry.

Extends: oval-def:StateType

Child Elements

Table 912: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
domain_name	oval-def:EntityStateStringType (0..1)	The domain_name element contains a string that represents a domain name that was collected from the DNS cache on the local system.
ttl	oval-def:EntityStateIntType (0..1)	The ttl element contains an integer that represents the time to live in seconds of the DNS cache entry.
ip_address	oval-def:EntityStateIPAddressStringType (0..1)	The ip_address element contains a string that represents an IP address associated with the specified domain name that was collected from the DNS cache on the local system. Note that the IP address can be IPv4 or IPv6.

< file_test >

The file test is used to check metadata associated with UNIX files, of the sort returned by either an ls command, stat command or stat() system call. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a file_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 913: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< file_object >

The file_object element is used by a file test to define the specific file(s) to be evaluated. The file_object will collect all UNIX file types (directory, regular file, character device, block device, fifo, symbolic link, and socket). Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A file object defines the path and filename of the file(s). In addition, a number of behaviors may be provided that help guide the collection of objects. Please refer to the FileBehaviors complex type for more information about specific behaviors.

The set of files to be evaluated may be identified with either a complete filepath or a path and filename. Only one of these options may be selected.

It is important to note that 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. This is because the ‘filepath’ element represents an absolute path to a particular file and it is not possible to recurse over a file.

Extends: oval-def:ObjectType

Child Elements

Table 914: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	unix-def:FileBehaviors (0..1)	
filepath	oval-def:EntityObjectStringType (1..1)	The filepath element specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath.
path	oval-def:EntityObjectStringType (1..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-def:EntityObjectStringType (1..1)	The filename element specifies the name of a file to evaluate. If the xsi:nil attribute is set to true, then the object being specified is the higher level directory object (not all the files in the directory). In this case, the filename element should not be used during collection and would result in the unique set of items being the directories themselves. For example, one would set xsi:nil to true if the desire was to test the attributes or permissions associated with a directory. Setting xsi:nil equal to true is different than using a .* pattern match, which says to collect every file under a given path.
oval-def:filter	n/a (0..unbounded)	

< file_state >

The file_state element defines the different metadata associate with a UNIX file. This includes the path, filename, type, group id, user id, size, etc. In addition, the permission associated with the file are also included. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 915: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-def:EntityStateStringType (0..1)	The filepath element specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath.
path	oval-def:EntityStateStringType (0..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-def:EntityStateStringType (0..1)	The name of the file.
type	oval-def:EntityStateStringType (0..1)	This is the file's type: regular file (regular), directory, named pipe (fifo), symbolic link, socket or block special.
group_id	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	The group_id entity represents the group owner of a file, by group number.
user_id	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	The numeric user id, or uid, is the third column of each user's entry in /etc/passwd. This element represents the owner of the file.
a_time	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	This is the time that the file was last accessed, in seconds since the Unix epoch. The Unix epoch is the time 00:00:00 UTC on January 1, 1970.
c_time	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	This is the time of the last change to the file's inode, in seconds since the Unix epoch. The Unix epoch is the time 00:00:00 UTC on January 1, 1970. An inode is a Unix data structure that stores all of the information about a particular file.
m_time	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	This is the time of the last change to the file's contents, in seconds since the Unix epoch. The Unix epoch is the time 00:00:00 UTC on January 1, 1970.
size	oval-def:EntityStateIntType (0..1)	This is the size of the file in bytes.
suid	oval-def:EntityStateBoolType (0..1)	Does the program run with the uid (thus privileges) of the file's owner, rather than the calling user?
sgid	oval-def:EntityStateBoolType (0..1)	Does the program run with the gid (thus privileges) of the file's group owner, rather than the calling user's group?
sticky	oval-def:EntityStateBoolType (0..1)	Can users delete each other's files in this directory, when said directory is writable by those users?
uread	oval-def:EntityStateBoolType (0..1)	Can the owner (user owner) of the file read this file or, if a directory, read the directory contents?
uwrite	oval-def:EntityStateBoolType (0..1)	Can the owner (user owner) of the file write to this file or, if a directory, write to the directory?
uexec	oval-def:EntityStateBoolType (0..1)	Can the owner (user owner) of the file execute it or, if a directory, change into the directory?

== FileBehaviors ==

The FileBehaviors complex type defines a number of behaviors that allow a more detailed definition of the file_object being specified. 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.

It is important to note that 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. This is because the 'filepath' element represents an absolute path to a particular file and it is not possible to recurse over a file.

Attributes

Table 916: Attributes

Attribute	Type	Desc.
max_depth	Restriction of xsd:integer (optional *default*='-1')	'max_depth' defines the maximum depth of recursion to perform when a recurse_direction is specified. A value of '0' is equivalent to no recursion, '1' means to step only one directory level up/down, and so on. The default value is '-1' meaning no limitation. For a 'max_depth' of -1 or any value of 1 or more the starting directory must be considered in the recursive search.

Note that the default recurse_direction behavior is 'none' so even though max_depth specifies no limitation by default, the recurse_direction behavior turns recursion off. Note that this behavior only applies with the equality operation on the path entity.

- - recurse
  - Restriction of xsd:string (optional *default*='symlinks and directories') ('~~none~~', '~~files~~', 'directories', '~~files and directories~~', 'symlinks', 'symlinks and directories')
  - 'recurse' defines how to recurse into the path entity, in other words what to follow during recursion. Options include symlinks, directories, or both. Note that a max-depth other than 0 has to be specified for recursion to take place and for this attribute to mean anything.

Note that this behavior only applies with the equality operation on the path entity.

- - recurse_direction
  - Restriction of xsd:string (optional *default*='none') ('none', 'up', 'down')
  - 'recurse_direction' defines the direction to recurse, either 'up' to parent directories, or 'down' into child directories. The default value is 'none' for no recursion.

Note that this behavior only applies with the equality operation on the path entity.

- - recurse_file_system
  - Restriction of xsd:string (optional *default*='all') ('all', 'local', 'defined')
  - 'recurse_file_system' defines the file system limitation of any searching and applies to all operations as specified on the path or filepath entity. The value of 'local' limits the search scope to local file systems (as opposed to file systems mounted from an external system). The value of 'defined' keeps any recursion within the file system that the file_object (path+filename or filepath) has specified. For example, if the path specified was "/", you would search only the filesystem mounted there, not other filesystems mounted to descendant paths. The value of 'defined' only applies when an equality

operation is used for searching because the path or filepath entity must explicitly define a file system. The default value is 'all' meaning to search all available file systems for data collection.

Note that 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. Searching 'all' file systems may have performance implications.

< fileextendedattribute_test >

The file extended attribute test is used to check extended attribute values associated with UNIX files, of the sort returned by the getfattr command or getxattr() system call. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a fileextendedattribute_object and the optional state element specifies the extended attributes to check.

NOTE: Solaris has a very different implementation of "extended attributes" in which the attributes are really an orthogonal directory hierarchy of files. See the Solaris documentation for more details. The file extended attribute test only handles simple name/value pairs as implemented by most other UNIX derived operating systems.

Extends: oval-def:TestType

Child Elements

Table 917: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< fileextendedattribute_object >

The fileextendedattribute_object element is used by a file extended attribute test to define the specific file(s) and attribute(s) to be evaluated. The fileextendedattribute_object will collect all UNIX file types (directory, regular file, character device, block device, fifo, symbolic link, and socket). Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A file extended attribute object defines the path, filename and attribute name. In addition, a number of behaviors may be provided that help guide the collection of objects. Please refer to the FileExtendedAttributeBehaviors complex type for more information about specific behaviors.

The set of files to be evaluated may be identified with either a complete filepath or a path and filename. Only one of these options may be selected.

It is important to note that 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. This is because the 'filepath' element represents an absolute path to a particular file and it is not possible to recurse over a file.

Extends: oval-def:ObjectType

Child Elements

Table 918: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	unix-def:FileBehaviors (0..1)	
filepath	oval-def:EntityObjectStringType (1..1)	The filepath element specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath.
path	oval-def:EntityObjectStringType (1..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-def:EntityObjectStringType (1..1)	The filename element specifies the name of a file to evaluate. If the xsi:nil attribute is set to true, then the object being specified is the higher level directory object (not all the files in the directory). In this case, the filename element should not be used during collection and would result in the unique set of items being the directories themselves. For example, one would set xsi:nil to true if the desire was to test the attributes associated with a directory. Setting xsi:nil equal to true is different than using a .* pattern match, which says to collect every file under a given path.
attribute_name	oval-def:EntityObjectStringType (1..1)	The attribute_name element specifies the name of an extended attribute to evaluate.
oval-def:filter	n/a (0..unbounded)	

< fileextendedattribute_state >

The fileextendedattribute_state element defines an extended attribute associated with a UNIX file. This includes the path, filename, attribute name, and attribute value.

Extends: oval-def:StateType

Child Elements

Table 919: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-def:EntityStateStringType (0..1)	The filepath element specifies the absolute path for a file on the machine. A directory can be specified as a filepath.
path	oval-def:EntityStateStringType (0..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-def:EntityStateStringType (0..1)	The name of the file.
attribute_name	oval-def:EntityStateStringType (0..1)	This is the extended attribute's name, identifier or key.
value	oval-def:EntityStateAnySimpleType (0..1)	The value entity represents the extended attribute's value or contents. To test for an attribute with no value assigned to it, this entity would be used with an empty value.

< gconf_test >

The gconf_test is used to check the attributes and value(s) associated with GConf preference keys. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a gconf_object and the optional gconf_state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 920: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< gconf_object >

The gconf_object element is used by a gconf_test to define the preference keys to collect and the sources from which to collect the preference keys. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 921: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
key	oval-def:EntityObjectStringType (1..1)	This is the preference key to check.
source	oval-def:EntityObjectStringType (1..1)	The source element specifies the source from which to collect the preference key. The source is represented by the absolute path to a GConf XML file as XML is the current backend for GConf. Note that other backends may become available in the future. If the xsi:nil attribute is set to 'true', the preference key is looked up using the GConf daemon. Otherwise, the preference key is looked up using the values specified in this entity.
oval-def:filter	n/a (0..unbounded)	

< gconf_state >

The gconf_state element defines the different information that can be used to evaluate the specified GConf preference key. This includes the preference key, source, type, whether it's writable, the user who last modified it, the time it was last modified, whether it's the default value, as well as the preference key's value. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 922: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
key	oval-def:EntityStateStringType (0..1)	The preference key to check.
source	oval-def:EntityStateStringType (0..1)	The source used to look up the preference key.
type	unix-def:EntityStateGconfTypeType (0..1)	The type of the preference key.
is_writable	oval-def:EntityStateBoolType (0..1)	Is the preference key writable? If true, the preference key is writable. If false, the preference key is not writable.
mod_user	oval-def:EntityStateStringType (0..1)	The user who last modified the preference key.
mod_time	oval-def:EntityStateIntType (0..1)	The time the preference key was last modified in seconds since the Unix epoch. The Unix epoch is the time 00:00:00 UTC on January 1, 1970.
is_default	oval-def:EntityStateBoolType (0..1)	Is the preference key value the default value. If true, the preference key value is the default value. If false, the preference key value is not the default value.
value	oval-def:EntityStateAnySimpleType (0..1)	The value of the preference key.

< inetd_test >

The inetd test is used to check information associated with different Internet services. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an inetd_object and the optional state element specifies the information to check.

Extends: oval-def:TestType

Child Elements

Table 923: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< inetd_object >

The inetd_object element is used by an inetd test to define the specific protocol-service to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An inetd object consists of a protocol entity and a service_name entity that identifies the specific service to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 924: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
protocol	oval-def:EntityObjectStringType (1..1)	A recognized protocol listed in the file /etc/inet/protocols.
service_name	oval-def:EntityObjectStringType (1..1)	The name of a valid service listed in the services file. For RPC services, the value of the service_name field consists of the RPC service name or program number, followed by a '/' (slash) and either a version number or a range of version numbers (for example, rstatd/2-4).
oval-def:filter	n/a (0..unbounded)	

< inetd_state >

The inetd_state element defines the different information associated with a specific Internet service. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 925: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
protocol	oval-def:EntityStateStringType (0..1)	A recognized protocol listed in the file /etc/inet/protocols.
service_name	oval-def:EntityStateStringType (0..1)	The name of a valid service listed in the services file. For RPC services, the value of the service-name field consists of the RPC service name or program number, followed by a '/' (slash) and either a version number or a range of version numbers (for example, rstatd/2-4).
server_program	oval-def:EntityStateStringType (0..1)	Either the pathname of a server program to be invoked by inetd to perform the requested service, or the value internal if inetd itself provides the service.
server_arguments	oval-def:EntityStateStringType (0..1)	The arguments for running the service. These are either passed to the server program invoked by inetd, or used to configure a service provided by inetd. In the case of server programs, the arguments shall begin with argv[0], which is typically the name of the program. In the case of a service provided by inted, the first argument shall be the word "internal".
endpoint_type	unix-def:EntityStateEndpointType (0..1)	The endpoint type (aka, socket type) associated with the service.
exec_as_user	oval-def:EntityStateStringType (0..1)	The user id of the user the server program should run under. (This allows for running with less permission than root.)
wait_status	unix-def:EntityStateWaitStatusType (0..1)	This field has values wait or nowait. This entry specifies whether the server that is invoked by inetd will take over the listening socket associated with the service, and whether once launched, inetd will wait for that server to exit, if ever, before it resumes listening for new service requests.

< interface_test >

The interface test enumerates various attributes about the interfaces on a system. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an interface_object and the optional state element specifies the interface information to check.

Extends: oval-def:TestType

Child Elements

Table 926: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< interface_object >

The interface_object element is used by an interface test to define the specific interfaces(s) to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An interface object consists of a single name entity that identifies which interface is being specified.

Extends: oval-def:ObjectType

Child Elements

Table 927: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityObjectStringType (1..1)	The name element is the interface (eth0, eth1, fw0, etc.) name to check.
oval-def:filter	n/a (0..unbounded)	

< interface_state >

The interface_state element enumerates the different properties associate with a Unix interface. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 928: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	The name element is the interface (eth0, eth1, fw0, etc.) name to check.
type	unix-def:EntityStateInterfaceType (0..1)	The type element specifies the type of interface.
hardware_addr	oval-def:EntityStateStringType (0..1)	The hardware_addr element is the hardware or MAC address of the physical network card. MAC addresses should be formatted according to the IEEE 802-2001 standard which states that a MAC address is a sequence of six octet values, separated by hyphens, where each octet is represented by two hexadecimal digits. Uppercase letters should also be used to represent the hexadecimal digits A through F.
inet_addr	oval-def:EntityStateIPAddressStringType (0..1)	This is the IP address of the interface. Note that the IP address can be IPv4 or IPv6. If the IP address is an IPv6 address, this entity will be expressed as an IPv6 address prefix using CIDR notation and the netmask entity will not be collected.
broadcast_addr	oval-def:EntityStateIPAddressStringType (0..1)	This is the broadcast IP address for this interface's network. Note that the IP address can be IPv4 or IPv6.
netmask	oval-def:EntityStateIPAddressStringType (0..1)	This is the bitmask used to calculate the interface's IP network. The network number is calculated by bitwise-ANDing this with the IP address. The host number on that network is calculated by bitwise-XORing this with the IP address. Note that if the inet_addr entity contains an IPv6 address prefix, this entity will not be collected.
flag	oval-def:EntityStateStringType (0..1)	The flag entity represents the interface flag line, which generally contains flags like "UP" to denote an active interface, "PROMISC" to note that the interface is listening for Ethernet frames not specifically addressed to it, and others. This element can be included multiple times in a system characteristic item in order to record a multitude of flags. Note that the entity_check attribute associated with EntityStateStringType guides the evaluation of entities like this that refer to items that can occur an unbounded number of times.

< password_test >

/etc/passwd. See passwd(4).

The password test is used to check metadata associated with the UNIX password file, of the sort returned by the passwd command. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a password_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 929: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< password_object >

The password_object element is used by a password test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A password object consists of a single username entity that identifies the user(s) whose password is to be evaluated.

Extends: oval-def:ObjectType

Child Elements

Table 930: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
username	oval-def:EntityObjectStringType (1..1)	The user(s) account whose password is to be evaluated.
oval-def:filter	n/a (0..unbounded)	

< password_state >

The password_state element defines the different information associated with the system passwords. Please refer to the individual elements in the schema for more details about what each represents.

See documentation on /etc/passwd for more details on the fields.

Extends: oval-def:StateType

Child Elements

Table 931: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
username	oval-def:EntityStateStringType (0..1)	The UNIX account name.
password	oval-def:EntityStateStringType (0..1)	This is the encrypted version of the user's password.
user_id	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	The numeric user id, or uid, is the third column of each user's entry in /etc/passwd.
group_id	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	The id of the primary UNIX group the user belongs to.
gcos	oval-def:EntityStateStringType (0..1)	The GECOS (or GCOS) field from /etc/passwd; typically contains the user's full name.
home_dir	oval-def:EntityStateStringType (0..1)	The user's home directory.
login_shell	oval-def:EntityStateStringType (0..1)	The user's shell program.
last_login	oval-def:EntityStateIntType (0..1)	The date and time when the last login occurred. This value is stored as the number of seconds that have elapsed since 00:00:00, January 1, 1970, UTC.

< process_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.8
- Reason: The process_test has been deprecated and replaced by the process58_test. The command line of a process cannot be used to uniquely identify a process. As a result, the pid entity was added to the process58_object. Please see the process58_test for additional information.

The process test is used to check information found in the UNIX processes. It is equivalent to parsing the output of the ps command. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a process_object and the optional state element specifies the process information to check.

Extends: oval-def:TestType

Child Elements

Table 932: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< process_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.8
- Reason: The process_object has been deprecated and replaced by the process58_object. The command line of a process cannot be used to uniquely identify a process. As a result, the pid entity was added to the process58_object. Please see the process58_object for additional information.

The process_object element is used by a process test to define the specific process(es) to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A process object defines the command line used to start the process(es).

Extends: oval-def:ObjectType

Child Elements

Table 933: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
command	oval-def:EntityObjectStringType (1..1)	The command element specifies the command/program name to check.

< process_state > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.8
- Reason: The process_state has been deprecated and replaced by the process58_state. The command line of a process cannot be used to uniquely identify a process. As a result, the pid entity was added to the process58_object. Please see the process58_state for additional information.

The process_state element defines the different metadata associated with a UNIX process. This includes the command line, pid, ppid, priority, and user id. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 934: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
command	oval-def:EntityStateStringType (0..1)	The command element specifies the command/program name to check.
exec_time	oval-def:EntityStateStringType (0..1)	This is the cumulative CPU time, formatted in [DD-]HH:MM:SS where DD is the number of days when execution time is 24 hours or more.
pid	oval-def:EntityStateIntType (0..1)	This is the process ID of the process.
ppid	oval-def:EntityStateIntType (0..1)	This is the process ID of the process's parent process.
priority	oval-def:EntityStateIntType (0..1)	This is the scheduling priority with which the process runs. This can be adjusted with the nice command or nice() system call.
ruid	oval-def:EntityStateIntType (0..1)	This is the real user id which represents the user who has created the process.
scheduling_class	oval-def:EntityStateStringType (0..1)	A platform specific characteristic maintained by the scheduler: RT (real-time), TS (timeshare), FF (fifo), SYS (system), etc.
start_time	oval-def:EntityStateStringType (0..1)	This is the time of day the process started formatted in HH:MM:SS if the same day the process started or formatted as MMM_DD (Ex.: Feb_5) if process started the previous day or further in the past.
tty	oval-def:EntityStateStringType (0..1)	This is the TTY on which the process was started, if applicable.
user_id	oval-def:EntityStateIntType (0..1)	This is the effective user id which represents the actual privileges of the process.

< process58_test >

The process58_test is used to check information found in the UNIX processes. It is equivalent to parsing the output of the ps command. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a process58_object and the optional state element references a process58_state that specifies the process information to check.

Extends: oval-def:TestType

Child Elements

Table 935: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< process58_object >

The process58_object element is used by a process58_test to define the specific process(es) to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A process58_object defines the command line used to start the process(es) and pid.

Extends: oval-def:ObjectType

Child Elements

Table 936: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
command_line	oval-def:EntityObjectStringType (1..1)	The command_line entity is the string used to start the process. This includes any parameters that are part of the command line.
pid	oval-def:EntityObjectIntType (1..1)	The pid entity is the process ID of the process.
oval-def:filter	n/a (0..unbounded)	

< process58_state >

The process58_state element defines the different metadata associated with a UNIX process. This includes the command line, pid, ppid, priority, and user id. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 937: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
command_line	oval-def:EntityStateStringType (0..1)	This is the string used to start the process. This includes any parameters that are part of the command line.
exec_time	oval-def:EntityStateStringType (0..1)	This is the cumulative CPU time, formatted in [DD-]HH:MM:SS where DD is the number of days when execution time is 24 hours or more.
pid	oval-def:EntityStateIntType (0..1)	This is the process ID of the process.
ppid	oval-def:EntityStateIntType (0..1)	This is the process ID of the process's parent process.
priority	oval-def:EntityStateIntType (0..1)	This is the scheduling priority with which the process runs. This can be adjusted with the nice command or nice() system call.
ruid	oval-def:EntityStateIntType (0..1)	This is the real user id which represents the user who has created the process.
scheduling_class	oval-def:EntityStateStringType (0..1)	A platform specific characteristic maintained by the scheduler: RT (real-time), TS (timeshare), FF (fifo), SYS (system), etc.
start_time	oval-def:EntityStateStringType (0..1)	This is the time of day the process started formatted in HH:MM:SS if the same day the process started or formatted as MMM_DD (Ex.: Feb_5) if process started the previous day or further in the past.
tty	oval-def:EntityStateStringType (0..1)	This is the TTY on which the process was started, if applicable.
user_id	oval-def:EntityStateIntType (0..1)	This is the effective user id which represents the actual privileges of the process.
exec_shield	oval-def:EntityStateBoolType (0..1)	A boolean that when true would indicates that ExecShield is enabled for the process. Applicable only to RedHat-based Linux distros, an example script demonstrating the collection of this entity can be found at http://people.redhat.com/sgrubb/files/lsexec
loginuid	oval-def:EntityStateIntType (0..1)	The loginuid shows which account a user gained access to the system with. The /proc/XXXX/loginuid shows this value.
posix_capability	oval-def:EntityStateCapabilityType (0..1)	An effective capability associated with the process. See linux/include/linux/capability.h for more information.
selinux_domain_label	oval-def:EntityStateStringType (0..1)	An selinux domain label associated with the process.
session_id	oval-def:EntityStateIntType (0..1)	The session ID of the process.

< routingtable_test >

The routingtable_test is used to check information about the IPv4 and IPv6 routing table entries found in a system's primary routing table. It is important to note that only numerical addresses will be collected and that their symbolic representations will not be resolved. This equivalent to using the '-n' option with route(8) or netstat(8). It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a routingtable_object and the optional routingtable_state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 938: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< routingtable_object >

The routingtable_object element is used by a routingtable_test to define the destination IP address(es), found in a system's primary routing table, to collect. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 939: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
destination	oval-def:EntityObjectIPAddressType (1..1)	This is the destination IP address of the routing table entry to check.
oval-def:filter	n/a (0..unbounded)	

< routingtable_state >

The routingtable_state element defines the different information that can be used to check an entry found in a system's primary routing table. This includes the destination IP address, gateway, netmask, flags, and the name of the interface associated with it. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 940: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
destination	oval-def:EntityStateIPAddressType (0..1)	The destination IP address prefix of the routing table entry. This is the destination IP address and netmask/prefix-length expressed using CIDR notation.
gateway	oval-def:EntityStateIPAddressType (0..1)	The gateway of the specified routing table entry.
flags	unix-def:EntityStateRoutingTableFlagsType (0..1)	The flags associated with the specified routing table entry.
interface_name	oval-def:EntityStateStringType (0..1)	The name of the interface associated with the routing table entry.

< runlevel_test >

The runlevel test is used to check information about which runlevel specified services are scheduled to exist at. For more information see the output generated by a chkconfig –list. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a runlevel_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 941: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< runlevel_object >

The runlevel_object element is used by a runlevel_test to define the specific service(s)/runlevel combination to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 942: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
service_name	oval-def:EntityObjectStringType (1..1)	The service_name entity refers to the name associated with a service. This name is usually the filename of the script file located in the /etc/init.d directory.
runlevel	oval-def:EntityObjectStringType (1..1)	The system runlevel to examine. A runlevel is defined as a software configuration of the system that allows only a selected group of processes to exist.
oval-def:filter	n/a (0..unbounded)	

< runlevel_state >

The runlevel_state element holds information about whether a specific service is scheduled to start or stop at a given runlevel. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 943: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
service_name	oval-def:EntityStateStringType (0..1)	The service_name entity refers the name associated with a service. This name is usually the filename of the script file located in the /etc/init.d directory.
runlevel	oval-def:EntityStateStringType (0..1)	The runlevel entity refers to the system runlevel associated with a service. A runlevel is defined as a software configuration of the system that allows only a selected group of processes to exist.
start	oval-def:EntityStateBoolType (0..1)	The start entity determines if the process is scheduled to be spawned at the specified runlevel.
kill	oval-def:EntityStateBoolType (0..1)	The kill entity determines if the process is supposed to be killed at the specified runlevel.

< sccs_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.10

- Reason: The sccs_test has been deprecated because the Source Code Control System (SCCS) is obsolete. The sccs_test may be removed in a future version of the language.

Extends: oval-def:TestType

Child Elements

Table 944: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< sccs_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.10
- Reason: The sccs_object has been deprecated because the Source Code Control System (SCCS) is obsolete. The sccs_object may be removed in a future version of the language.

The set of files to be evaluated may be identified with either a complete filepath or a path and filename. Only one of these options may be selected.

It is important to note that 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. This is because the ‘filepath’ element represents an absolute path to a particular file and it is not possible to recurse over a file.

Extends: oval-def:ObjectType

Child Elements

Table 945: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	unix-def:FileBehaviors (0..1)	
filepath	oval-def:EntityObjectStringType (1..1)	The filepath element specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath.
path	oval-def:EntityObjectStringType (1..1)	The path element specifies the directory component of the absolute path to an SCCS file.
filename	oval-def:EntityObjectStringType (1..1)	The name of an SCCS file.
oval-def:filter	n/a (0..unbounded)	

< sccs_state > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.10
- Reason: The sccs_state has been deprecated because the Source Code Control System (SCCS) is obsolete. The sccs_state may be removed in a future version of the language.

Extends: oval-def:StateType

Child Elements

Table 946: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-def:EntityStateStringType (0..1)	The filepath element specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath.
path	oval-def:EntityStateStringType (0..1)	The path element specifies the directory component of the absolute path to an SCCS file.
filename	oval-def:EntityStateStringType (0..1)	This is the name of a SCCS file.
module_name	oval-def:EntityStateStringType (0..1)	
module_type	oval-def:EntityStateStringType (0..1)	
release	oval-def:EntityStateStringType (0..1)	
level	oval-def:EntityStateStringType (0..1)	
branch	oval-def:EntityStateStringType (0..1)	
sequence	oval-def:EntityStateStringType (0..1)	
what_string	oval-def:EntityStateStringType (0..1)	

< shadow_test >

The shadow test is used to check information from the /etc/shadow file for a specific user. This file contains a user's password, but also their password aging and lockout information. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an shadow_object and the optional state element specifies the information to check.

Extends: oval-def:TestType

Child Elements

Table 947: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< shadow_object >

The shadow_object element is used by a shadow test to define the shadow file to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A shdow object consists of a single user entity that identifies the username associted with the shadow file.

Extends: oval-def:ObjectType

Child Elements

Table 948: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
username	oval-def:EntityObjectStringType (1..1)	
oval-def:filter	n/a (0..unbounded)	

< shadow_state >

The shadows_state element defines the different information associated with the system shadow file. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 949: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
username	oval-def:EntityStateStringType (0..1)	This is the name of the user being checked.
password	oval-def:EntityStateStringType (0..1)	This is the encrypted version of the user's password.
chg_lst	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	This is the date of the last password change in days since 1/1/1970.
chg_allow	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	This specifies how often in days a user may change their password. It can also be thought of as the minimum age of a password.
chg_req	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	This describes how long the user can keep a password before the system forces them to change it.
exp_warn	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	This describes how long before password expiration the system begins warning the user. The system will warn the user at each login.
exp_inact	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	The exp_inact entity describes how many days of account inactivity the system will wait after a password expires before locking the account. Unix systems are generally configured to only allow a given password to last for a fixed period of time. When this time, the chg_req parameter, is about to run out, the system begins warning the user at each login. How soon before the expiration the user receives these warnings is specified in exp_warn. The only hiccup in this design is that a user may not login in time to ever receive a warning before account expiration. The exp_inact parameter gives the sysadmin flexibility so that a user who reaches the end of their expiration time gains exp_inact more days to login and change their password manually.
exp_date	Restric-	This specifies when will the account's password expire, in days since 1/1/1970.

< symlink_test >

The symlink_test is used to obtain canonical path information for symbolic links.

Extends: oval-def:TestType

Child Elements

Table 950: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< symlink_object >

The symlink_object element is used by a symlink_test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A symlink_object consists of a filepath entity that contains the path to a symbolic link file. The resulting item identifies the canonical path of the link target (followed to its final destination, if there are intermediate links), an error if the link target does not exist or is a circular link (e.g., a link to itself). If the file located at filepath is not a symlink, or if there is no file located at the filepath, then any resulting item would itself have a status of does not exist.

Extends: oval-def:ObjectType

Child Elements

Table 951: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-def:EntityObjectStringType (1..1)	Specifies the filepath for the symbolic link.
oval-def:filter	n/a (0..unbounded)	

< symlink_state >

The symlink_state element defines a value used to evaluate the result of a specific symlink_object item.

Extends: oval-def:StateType

Child Elements

Table 952: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-def:EntityStateStringType (0..1)	Specifies the filepath used to create the object.
canonical_path	oval-def:EntityStateStringType (0..1)	Specifies the canonical path for the target of a symbolic link file specified by the filepath.

< sysctl_test >

The sysctl_test is used to check the values associated with the kernel parameters that are used by the local system. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a sysctl_object and the optional state element references a sysctl_state that specifies the information to check.

Extends: oval-def:TestType

Child Elements

Table 953: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< sysctl_object >

The sysctl_object is used by a sysctl_test to define which kernel parameters on the local system should be collected. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 954: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityObjectStringType (1..1)	The name element specifies the name(s) of the kernel parameter(s) that should be collected from the local system.
oval-def:filter	n/a (0..unbounded)	

< sysctl_state >

The sysctl_state contains two entities that are used to check the kernel parameter name and value(s).

Extends: oval-def:StateType

Child Elements

Table 955: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	The name element contains a string that represents the name of a kernel parameter that was collected from the local system.
value	oval-def:EntityStateAnySimpleType (0..1)	The value element contains a string that represents the value(s) associated with the specified kernel parameter.

< uname_test >

The uname test reveals information about the hardware the machine is running on. This information is the parsed equivalent of uname -a. For example: “Linux quark 2.6.5-7.108-default #1 Wed Aug 25 13:34:40 UTC 2004 i686 i686 i386 GNU/Linux” or “Darwin TestHost 7.7.0 Darwin Kernel Version 7.7.0: Sun Nov 7 16:06:51 PST 2004; root:xnu/xnu-517.9.5.obj~1/RELEASE_PPC Power Macintosh powerpc”. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a uname_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 956: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< uname_object >

The uname_object element is used by an uname test to define those objects to evaluated based on a specified state. There is actually only one object relating to uname and this is the system as a whole. Therefore, there are no child entities defined. Any OVAL Test written to check uname will reference the same uname_object which is basically an empty object element.

Extends: oval-def:ObjectType

< uname_state >

The uname_state element defines the information about the hardware the machine is running one. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 957: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
machine_class	oval-def:EntityStateStringType (0..1)	This entity specifies a machine hardware name. This corresponds to the command uname -m.
node_name	oval-def:EntityStateStringType (0..1)	This entity specifies a host name. This corresponds to the command uname -n.
os_name	oval-def:EntityStateStringType (0..1)	This entity specifies an operating system name. This corresponds to the command uname -s.
os_release	oval-def:EntityStateStringType (0..1)	This entity specifies a build version. This corresponds to the command uname -r.
os_version	oval-def:EntityStateStringType (0..1)	This entity specifies an operating system version. This corresponds to the command uname -v.
processor_type	oval-def:EntityStateStringType (0..1)	This entity specifies a processor type. This corresponds to the command uname -p.

< xinetd_test >

The xinetd test is used to check information associated with different Internet services. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an inetd_object and the optional state element specifies the information to check.

Extends: oval-def:TestType

Child Elements

Table 958: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< xinetd_object >

The xinetd_object element is used by an xinetd test to define the specific protocol-service to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An xinetd object consists of a protocol entity and a service_name entity that identifies the specific service to be tested.

Extends: oval-def:ObjectType

Child Elements

Table 959: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
protocol	oval-def:EntityObjectStringType (1..1)	The protocol entity specifies the protocol that is used by the service. The list of valid protocols can be found in /etc/protocols.
service_name	oval-def:EntityObjectStringType (1..1)	The service_name entity specifies the name of the service.
oval-def:filter	n/a (0..unbounded)	

< xinetd_state >

The xinetd_state element defines the different information associated with a specific Internet service. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 960: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
protocol	oval-def:EntityStateStringType (0..1)	The protocol entity specifies the protocol that is used by the service. The list of valid protocols can be found in /etc/protocols.
service_name	oval-def:EntityStateStringType (0..1)	The service_name entity specifies the name of the service.
flags	oval-def:EntityStateStringType (0..1)	The flags entity specifies miscellaneous settings associated with the service.
no_access	oval-def:EntityStateStringType (0..1)	The no_access entity specifies the remote hosts to which the service is unavailable. Please see the xinetd.conf(5) man page for information on the different formats that can be used to describe a host.
only_from	oval-def:EntityStateIPAddressStringType (0..1)	The only_from entity specifies the remote hosts to which the service is available. Please see the xinetd.conf(5) man page for information on the different formats that can be used to describe a host.
port	oval-def:EntityStateIntType (0..1)	The port entity specifies the port used by the service.
server	oval-def:EntityStateStringType (0..1)	The server entity specifies the executable that is used to launch the service.
server_arguments	oval-def:EntityStateStringType (0..1)	The server_arguments entity specifies the arguments that are passed to the executable when starting the service.
socket_type	oval-def:EntityStateStringType (0..1)	The socket_type entity specifies the type of socket that is used by the service. Possible values include stream, dgram, raw, or seqpacket.
type	unix-def:EntityStateXinetdTypeStatusType (0..1)	The type entity specifies the type of the service. A service may have multiple types.
user	oval-def:EntityStateStringType (0..1)	The user entity specifies the user identifier of the process that is running the service. The user identifier may be expressed as a numerical value or as a user name that exists in /etc/passwd.
wait	oval-def:EntityStateBoolType (0..1)	The wait entity specifies whether or not the service is single-threaded or multi-threaded and whether or not xinetd accepts the connection or the service accepts the connection. A value of 'true' indicates that the service is single-threaded and the service will accept the connection. A value of 'false' indicates that the service is multi-threaded and xinetd will accept the connection.
disabled	oval-def:EntityStateBoolType (0..1)	The disabled entity specifies whether or not the service is disabled. A value of 'true' indicates that the service is disabled and will not start. A value of 'false' indicates that the service is not disabled.

== EntityStateCapabilityType ==

The EntityStateCapabilityType complex type restricts a string value to a specific set of values that describe POSIX capability types associated with a process service. This list is based off the values defined in linux/include/linux/capability.h. Documentation on each allowed value can be found in capability.h. The empty string is also allowed to support empty elements associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 961: Enumeration Values

Value	Description
CAP_CHOWN	
CAP_DAC_OVERRIDE	
CAP_DAC_READ_SEARCH	
CAP_FOWNER	
CAP_FSETID	
CAP_KILL	
CAP_SETGID	
CAP_SETUID	
CAP_SETPCAP	
CAP_LINUX_IMMUTABLE	
CAP_NET_BIND_SERVICE	
CAP_NET_BROADCAST	
CAP_NET_ADMIN	

Continued on next page

Table 961 – continued from previous page

Value	Description
CAP_NET_RAW	
CAP_IPC_LOCK	
CAP_IPC_OWNER	
CAP_SYS_MODULE	
CAP_SYS_RAWIO	
CAP_SYS_CHROOT	
CAP_SYS_PTRACE	
CAP_SYS_ADMIN	
CAP_SYS_BOOT	
CAP_SYS_NICE	
CAP_SYS_RESOURCE	
CAP_SYS_TIME	
CAP_SYS_TTY_CONFIG	
CAP_MKNOD	
CAP_LEASE	
CAP_AUDIT_WRITE	

Continued on next page

Table 961 – continued from previous page

Value	Description
CAP_AUDIT_CONTROL	
CAP_SETFCAP	
CAP_MAC_OVERRIDE	
CAP_MAC_ADMIN	
CAP_SYS_PACCT	
CAP_SYSLOG	
CAP_WAKE_ALARM	
CAP_BLOCK_SUSPEND	
CAP_AUDIT_READ	
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateEndpointType ==

The EntityStateEndpointType complex type restricts a string value to a specific set of values that describe endpoint types associated with an Internet service. The empty string is also allowed to support empty elements associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 962: Enumeration Values

Value	Description
stream	The stream value is used to describe a stream socket.
dgram	The dgram value is used to describe a datagram socket.
raw	The raw value is used to describe a raw socket.
seqpacket	The seqpacket value is used to describe a sequenced packet socket.
tli	The tli value is used to describe all TLI endpoints.
sunrpc_tcp	The sunrpc_tcp value is used to describe all SUNRPC TCP endpoints.
sunrpc_udp	The sunrpc_udp value is used to describe all SUNRPC UDP endpoints.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateGconfTypeType ==

The EntityStateGconfTypeType complex type restricts a string value to the seven values GCONF_VALUE_STRING, GCONF_VALUE_INT, GCONF_VALUE_FLOAT, GCONF_VALUE_BOOL, GCONF_VALUE_SCHEMA, GCONF_VALUE_LIST, and GCONF_VALUE_PAIR that specify the datatype of the value associated with a GConf preference key. The empty string is also allowed to support empty elements associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 963: Enumeration Values

Value	Description
GCONF_VALUE_STRING	The GCONF_VALUE_STRING type is used to describe a preference key that has a string value.
GCONF_VALUE_INT	The GCONF_VALUE_INT type is used to describe a preference key that has a integer value.
GCONF_VALUE_FLOAT	The GCONF_VALUE_FLOAT type is used to describe a preference key that has a float value.
GCONF_VALUE_BOOL	The GCONF_VALUE_BOOL type is used to describe a preference key that has a boolean value.
GCONF_VALUE_SCHEMA	The GCONF_VALUE_SCHEMA type is used to describe a preference key that has a schema value. The actual value will be the default value as specified in the GConf schema.
GCONF_VALUE_LIST	The GCONF_VALUE_LIST type is used to describe a preference key that has a list of values. The actual values will be one of the primitive GConf datatypes GCONF_VALUE_STRING, GCONF_VALUE_INT, GCONF_VALUE_FLOAT, GCONF_VALUE_BOOL, and GCONF_VALUE_SCHEMA. Note that all of the values associated with a GCONF_VALUE_LIST are required to have the same type.
GCONF_VALUE_PAIR	The GCONF_VALUE_PAIR type is used to describe a preference key that has a pair of values. The actual values will consist of the primitive GConf datatypes GCONF_VALUE_STRING, GCONF_VALUE_INT, GCONF_VALUE_FLOAT, GCONF_VALUE_BOOL, and GCONF_VALUE_SCHEMA. Note that the values associated with a GCONF_VALUE_PAIR are not required to have the same type.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateRoutingTableFlagsType ==

The EntityStateRoutingTableFlagsType complex type restricts a string value to a specific set of values that describe the flags associated with a routing table entry. This list is based off the values defined in the man pages of various platforms. For Linux, please see route(8). For Solaris, please see netstat(1M). For HP-UX, please see netstat(1). For Mac OS, please see netstat(1). For FreeBSD, please see netstat(1). Documentation on each allowed value can be found in the previously listed man pages. The empty string is also allowed to support empty elements associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 964: Enumeration Values

Value	Description
UP	
GATEWAY	
HOST	
REINSTATE	
DYNAMIC	
MODIFIED	
ADDRCONF	
CACHE	
REJECT	
REDUNDANT	
SETSRC	
BROADCAST	
LOCAL	
PROTOCOL_1	
PROTOCOL_2	
PROTOCOL_3	
BLACK_HOLE	
CLONING	
PROTOCOL_CLONING	

The following table is a mapping between the generic flag enumeration values and the actual flag values found on the various platforms. If the flag value is not specified, for a particular generic flag enumeration value, the flag value is not defined for that platform. ` Name Linux Solaris HPUX Mac OS FreeBSD   AIX UP U U U U
U   U GATEWAY G G G G G   G HOST H H H H H   H REINSTATE
R DYNAMIC D D D D   D MODIFIED M M M   M ADDRCONF A A CACHE C
e REJECT ! R R   R REDUNDANT M (>=9) SETSRC S BROADCAST B b
b   b LOCAL L   l PROTOCOL_1 1 1   1 PROTOCOL_2 2
2   2 PROTOCOL_3 3 3   3 BLACK_HOLE B B CLONING C C   c
PROTOCOL_CLONING c c INTERFACE_SCOPE I LINK_LAYER L L   L MULTICAST
m   m STATIC S S   S WAS_CLONED W W   W XRESOLVE X
X USABLE   u PINNED
P ACTIVE_DEAD_GATEWAY_DETECTION   A
(>=5.1) `

== EntityStateXinetdTypeStatusType ==

The EntityStateXinetdTypeStatusType complex type restricts a string value to five values, either RPC, INTERNAL, UNLISTED, TCPMUX, or TCPMUXPLUS that specify the type of service registered in xinetd. The empty string is also allowed to support empty elements associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 965: Enumeration Values

Value	Description
INTERNAL	The INTERNAL type is used to describe services like echo, chargen, and others whose functionality is supplied by xinetd itself.
RPC	The RPC type is used to describe services that use remote procedure call ala NFS.
UNLISTED	The UNLISTED type is used to describe services that aren't listed in /etc/protocols or /etc/rpc.
TCPMUX	The TCPMUX type is used to describe services that conform to RFC 1078. This type indiciates that the service is responsible for handling the protocol handshake.
TCPMUXPLUS	The TCPMUXPLUS type is used to describe services that conform to RFC 1078. This type indicates that xinetd is responsible for handling the protocol handshake.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateWaitStatusType ==

The EntityStateWaitStatusType complex type restricts a string value to two values, either wait or nowait, that specify whether the server that is invoked by inetd will take over the listening socket associated with the service, and whether once launched, inetd will wait for that server to exit, if ever, before it resumes listening for new service requests. The empty string is also allowed to support empty elements associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 966: Enumeration Values

Value	Description
wait	The value of ‘wait’ specifies that the server that is invoked by inetd will take over the listening socket associated with the service, and once launched, inetd will wait for that server to exit, if ever, before it resumes listening for new service requests.
nowait	The value of ‘nowait’ specifies that the server that is invoked by inetd will not wait for any existing server to finish before taking over the listening socket associated with the service.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateEncryptMethodType ==

The EntityStateEncryptMethodType complex type restricts a string value to a set that corresponds to the allowed encrypt methods used for protected passwords in a shadow file. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 967: Enumeration Values

Value	Description
DES	The DES method corresponds to the (none) prefix.
BSDi	The BSDi method corresponds to BSDi modified DES or the ‘_’ prefix.
MD5	The MD5 method corresponds to MD5 for Linux/BSD or the 1 prefix.
Blowfish	The Blowfish method corresponds to Blowfish (OpenBSD) or the 2 or $2a$ prefixes.
Sun MD5	The Sun MD5 method corresponds to the $md5$ prefix.
SHA-256	The SHA-256 method corresponds to the 5 prefix.
SHA-512	The SHA-512 method corresponds to the 6 prefix.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateInterfaceType ==

The EntityStateInterfaceType complex type restricts a string value to a specific set of values. These values describe the different interface types which are defined in ‘if_arp.h’. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 968: Enumeration Values

Value	Description
ARPHRD_ETHER	The ARPHRD_ETHER type is used to describe ethernet interfaces.
ARPHRD_FDDI	The ARPHRD_FDDI type is used to describe fiber distributed data interfaces (FDDI).
ARPHRD_LOOPBACK	The ARPHRD_LOOPBACK type is used to describe loopback interfaces.
ARPHRD_VOID	The ARPHRD_VOID type is used to describe unknown interfaces.
ARPHRD_PPP	The ARPHRD_PPP type is used to describe point-to-point protocol interfaces (PPP).
ARPHRD_SLIP	The ARPHRD_SLIP type is used to describe serial line internet protocol interfaces (SLIP).
ARPHRD_PRONET	The ARPHRD_PRONET type is used to describe PROnet token ring interfaces.
	The empty string value is permitted here to allow for empty elements associated with variable references.

Open Vulnerability and Assessment Language: Unix System Characteristics

- Schema: Unix System Characteristics
- Version: 5.11.1:1.2
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the UNIX specific system characteristic items found in Open Vulnerability and Assessment Language (OVAL). Each item is an extension of the standard item element defined in the Core System Characteristic Schema. Through extension, each item inherits a set of elements and attributes that are shared amongst all OVAL Items. Each item is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core System Characteristic Schema is not outlined here.

The OVAL Schema is maintained by the 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.cisecurity.org.

Item Listing

- *< dnscache_item >*
- *< file_item >*
- *< fileextendedattribute_item >*
- *< gconf_item >*
- *< inetd_item >*
- *< interface_item >*
- *< password_item >*
- *< process_item >*
- *< process58_item >*
- *< routingtable_item >*
- *< runlevel_item >*
- *< sccs_item > (Deprecated)*
- *< shadow_item >*
- *< symlink_item >*
- *< sysctl_item >*
- *< uname_item >*
- *< xinetd_item >*

< dnscache_item >

The dnscache_item stores information retrieved from the DNS cache about a domain name, its time to live, and its corresponding IP addresses.

Extends: oval-sc:ItemType

Child Elements

Table 969: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
domain_name	oval-sc:EntityItemStringType (0..1)	The domain_name element contains a string that represents a domain name that was collected from the DNS cache on the local system.
ttl	oval-sc:EntityItemIntType (0..1)	The ttl element contains an integer that represents the time to live in seconds of the DNS cache entry.
ip_address	oval-sc:EntityItemIPAddressStringType (0..unbounded)	The ip_address element contains a string that represents an IP address associated with the specified domain name. Note that the IP address can be IPv4 or IPv6.

< file_item >

The file item holds information about the individual files found on a system. Each file item contains path and filename information as well as its type, associated user and group ids, relevant dates, and the privialeges granted. It extends the standard ItemType as defined in the oval-system-characteristics schema and one should refer to the ItemType description for more information.

Extends: oval-sc:ItemType

Child Elements

Table 970: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-sc:EntityItemStringType (0..1)	The filepath element specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath.
path	oval-sc:EntityItemStringType (0..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-sc:EntityItemStringType (0..1)	The name of the file. If the xsi:nil attribute is set to true, then the item being represented is the higher directory represented by the path entity.
type	oval-sc:EntityItemStringType (0..1)	This is the file's type: regular file (regular), directory, named pipe (fifo), symbolic link, socket or block special.
group_id	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	This is the group owner of the file, by group number.
user_id	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	The numeric user id, or uid, is the third column of each user's entry in /etc/passwd. This element represents the owner of the file.
a_time	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	This is the time that the file was last accessed, in seconds since the Unix epoch. The Unix epoch is the time 00:00:00 UTC on January 1, 1970.
c_time	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	This is the time of the last change to the file's inode, in seconds since the Unix epoch. The Unix epoch is the time 00:00:00 UTC on January 1, 1970. An inode is a Unix data structure that stores all of the information about a particular file.
m_time	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	This is the time of the last change to the file's contents, in seconds since the Unix epoch. The Unix epoch is the time 00:00:00 UTC on January 1, 1970.
size	oval-sc:EntityItemIntType (0..1)	This is the size of the file in bytes.
suid	oval-sc:EntityItemBoolType (0..1)	Does the program run with the uid (thus privileges) of the file's owner, rather than the calling user?
sgid	oval-sc:EntityItemBoolType (0..1)	Does the program run with the gid (thus privileges) of the file's group owner, rather than the calling user's group?
sticky	oval-sc:EntityItemBoolType (0..1)	Can users delete each other's files in this directory, when said directory is writable by other users?
uread	oval-sc:EntityItemBoolType (0..1)	Can the owner (user owner) of the file read this file or, if a directory, read the directory contents?

< fileextendedattribute_item >

The file extended attribute item holds information about the individual file extended attributes found on a system. Each file extended attribute item contains path, filename, and attribute name information as well as the attribute's value. It extends the standard ItemType as defined in the oval-system-characteristics schema and one should refer to the ItemType description for more information.

Extends: oval-sc:ItemType

Child Elements

Table 971: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-sc:EntityItemStringType (0..1)	The filepath element specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath.
path	oval-sc:EntityItemStringType (0..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-sc:EntityItemStringType (0..1)	The name of the file. If the xsi:nil attribute is set to true, then the item being represented is the higher directory represented by the path entity.
attribute_name	oval-sc:EntityItemStringType (0..1)	This is the extended attribute's name, identifier or key.
value	oval-sc:EntityItemAnySimpleType (0..1)	This is the extended attribute's value or contents.

< gconf_item >

The gconf_item holds information about an individual GConf preference key found on a system. Each gconf_item contains a preference key, source, type, whether it's writable, the user who last modified it, the time it was last modified, whether it's the default value, as well as the preference key's value. It extends the standard ItemType as defined in the oval-system-characteristics schema and one should refer to the ItemType description for more information.

Extends: oval-sc:ItemType

Child Elements

Table 972: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
key	oval-sc:EntityItemStringType (0..1)	The preference key to check.
source	oval-sc:EntityItemStringType (0..1)	The source used to look up the preference key.
type	unix-sc:EntityItemGconfTypeType (0..1)	The type of the preference key.
is_writable	oval-sc:EntityItemBoolType (0..1)	Is the preference key writable? If true, the preference key is writable. If false, the preference key is not writable.
mod_user	oval-sc:EntityItemStringType (0..1)	The user who last modified the preference key.
mod_time	oval-sc:EntityItemIntType (0..1)	The time the preference key was last modified in seconds since the Unix epoch. The Unix epoch is the time 00:00:00 UTC on January 1, 1970.
is_default	oval-sc:EntityItemBoolType (0..1)	Is the preference key value the default value. If true, the preference key value is the default value. If false, the preference key value is not the default value.
value	oval-sc:EntityItemAnySimpleType (0..unbounded)	The value of the preference key.

< inetd_item >

The inetd item holds information associated with different Internet services. It extends the standard ItemType as defined in the oval-system-characteristics schema and one should refer to the ItemType description for more information.

Extends: oval-sc:ItemType

Child Elements

Table 973: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
protocol	oval-sc:EntityItemStringType (0..1)	A recognized protocol listed in the file /etc/inet/protocols.
service_name	oval-sc:EntityItemStringType (0..1)	The name of a valid service listed in the services file. For RPC services, the value of the service-name field consists of the RPC service name or program number, followed by a '/' (slash) and either a version number or a range of version numbers (for example, rstatd/2-4).
server_program	oval-sc:EntityItemStringType (0..1)	Either the pathname of a server program to be invoked by inetd to perform the requested service, or the value internal if inetd itself provides the service.
server_arguments	oval-sc:EntityItemStringType (0..1)	The arguments for running the service. These are either passed to the server program invoked by inetd, or used to configure a service provided by inetd. In the case of server programs, the arguments shall begin with argv[0], which is typically the name of the program. In the case of a service provided by inted, the first argument shall be the word "internal".
endpoint_type	unix-sc:EntityItemEndpointType (0..1)	The endpoint type (aka, socket type) associated with the service.
exec_as_user	oval-sc:EntityItemStringType (0..1)	The user id of the user the server program should run under. (This allows for running with less permission than root.)
wait_status	unix-sc:EntityItemWaitStatusType (0..1)	This field has values wait or nowait. This entry specifies whether the server that is invoked by inetd will take over the listening socket associated with the service, and whether once launched, inetd will wait for that server to exit, if ever, before it resumes listening for new service requests.

< interface_item >

The interface item holds information about the interfaces on a system. Each interface item contains name and address information as well as any associated flags. It extends the standard ItemType as defined in the oval-system-characteristics schema and one should refer to the ItemType description for more information.

Extends: oval-sc:ItemType

Child Elements

Table 974: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	The name entity is the actual name of the specific interface. Examples might be eth0, eth1, fwo, etc.
type	unix-sc:EntityItemInterfaceType (0..1)	This element specifies the type of interface.
hardware_addr	oval-sc:EntityItemStringType (0..1)	The hardware_addr entity is the hardware or MAC address of the physical network card. MAC addresses should be formatted according to the IEEE 802-2001 standard which states that a MAC address is a sequence of six octet values, separated by hyphens, where each octet is represented by two hexadecimal digits. Uppercase letters should also be used to represent the hexadecimal digits A through F.
inet_addr	oval-sc:EntityItemIPAddressStringType (0..1)	The inet_addr entity is the IP address of the specific interface. Note that the IP address can be IPv4 or IPv6. If the IP address is an IPv6 address, this entity should be expressed as an IPv6 address prefix using CIDR notation and the netmask entity should not be collected.
broadcast_addr	oval-sc:EntityItemIPAddressStringType (0..1)	The broadcast_addr entity is the broadcast IP address for this interface's network. Note that the IP address can be IPv4 or IPv6.
netmask	oval-sc:EntityItemIPAddressStringType (0..1)	This is the bitmask used to calculate the interface's IP network. The network number is calculated by bitwise-ANDing this with the IP address. The host number on that network is calculated by bitwise-XORing this with the IP address. Note that if the inet_addr entity contains an IPv6 address prefix, this entity should not be collected.
flag	oval-sc:EntityItemStringType (0..unbounded)	This is the interface flag line, which generally contains flags like "UP" to denote an active interface, "PROMISC" to note that the interface is listening for Ethernet frames not specifically addressed to it, and others.

< password_item >

/etc/passwd. See passwd(4).

Extends: oval-sc:ItemType

Child Elements

Table 975: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
username	oval-sc:EntityItemStringType (0..1)	This is the name of the user for which data was gathered.
password	oval-sc:EntityItemStringType (0..1)	This is the encrypted version of the user's password.
user_id	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	The numeric user id, or uid, is the third column of each user's entry in /etc/passwd.
group_id	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	The id of the primary UNIX group the user belongs to.
gcos	oval-sc:EntityItemStringType (0..1)	The GECOS (or GCOS) field from /etc/passwd; typically contains the user's full name.
home_dir	oval-sc:EntityItemStringType (0..1)	The user's home directory.
login_shell	oval-sc:EntityItemStringType (0..1)	The user's shell program.
last_login	oval-sc:EntityItemIntType (0..1)	The date and time when the last login occurred. This value is stored as the number of seconds that have elapsed since 00:00:00, January 1, 1970, UTC.

< process_item > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.8
- Reason: The process_item has been deprecated and replaced by the process58_item. The entity 'command' was changed to 'command_line' in the process58_item to accurately describe what information is collected. Please see the process58_item for additional information.

Output of /usr/bin/ps. See ps(1).

Extends: oval-sc:ItemType

Child Elements

Table 976: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
command	oval-sc:EntityItemStringType (0..1)	This specifies the command/program name about which data has has been collected.
exec_time	oval-sc:EntityItemStringType (0..1)	This is the cumulative CPU time, formatted in [DD-]HH:MM:SS where DD is the number of days when execution time is 24 hours or more.
pid	oval-sc:EntityItemIntType (0..1)	This is the process ID of the process.
ppid	oval-sc:EntityItemIntType (0..1)	This is the process ID of the process's parent process.
priority	oval-sc:EntityItemIntType (0..1)	This is the scheduling priority with which the process runs. This can be adjusted with the nice command or nice() system call.
ruid	oval-sc:EntityItemIntType (0..1)	This is the real user id which represents the user who has created the process.
scheduling_class	oval-sc:EntityItemStringType (0..1)	A platform specific characteristic maintained by the scheduler: RT (real-time), TS (timeshare), FF (fifo), SYS (system), etc.
start_time	oval-sc:EntityItemStringType (0..1)	This is the time of day the process started formatted in HH:MM:SS if the same day the process started or formatted as MMM_DD (Ex.: Feb_5) if process started the previous day or further in the past.
tty	oval-sc:EntityItemStringType (0..1)	This is the TTY on which the process was started, if applicable.
user_id	oval-sc:EntityItemIntType (0..1)	This is the effective user id which represents the actual privileges of the process.

< process58_item >

Output of /usr/bin/ps. See ps(1).

Extends: oval-sc:ItemType

Child Elements

Table 977: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
command_line	oval-sc:EntityItemStringType (0..1)	This is the string used to start the process. This includes any parameters that are part of the command line.
exec_time	oval-sc:EntityItemStringType (0..1)	This is the cumulative CPU time, formatted in [DD-]HH:MM:SS where DD is the number of days when execution time is 24 hours or more.
pid	oval-sc:EntityItemIntType (0..1)	This is the process ID of the process.
ppid	oval-sc:EntityItemIntType (0..1)	This is the process ID of the process's parent process.
priority	oval-sc:EntityItemIntType (0..1)	This is the scheduling priority with which the process runs. This can be adjusted with the nice command or nice() system call.
ruid	oval-sc:EntityItemIntType (0..1)	This is the real user id which represents the user who has created the process.
scheduling_class	oval-sc:EntityItemStringType (0..1)	A platform specific characteristic maintained by the scheduler: RT (real-time), TS (timeshare), FF (fifo), SYS (system), etc.
start_time	oval-sc:EntityItemStringType (0..1)	This is the time of day the process started formatted in HH:MM:SS if the same day the process started or formatted as MMM_DD (Ex.: Feb_5) if process started the previous day or further in the past.
tty	oval-sc:EntityItemStringType (0..1)	This is the TTY on which the process was started, if applicable.
user_id	oval-sc:EntityItemIntType (0..1)	This is the effective user id which represents the actual privileges of the process.
exec_shield	oval-sc:EntityItemBoolType (0..1)	A boolean that when true would indicates that ExecShield is enabled for the process.
loginuid	oval-sc:EntityItemIntType (0..1)	The loginuid shows which account a user gained access to the system with. The /proc/XXXX/loginuid shows this value.
posix_capability	unix-sc:EntityItemCapabilityType (0..unbounded)	An effective capability associated with the process. See linux/include/linux/capability.h for more information.
selinux_domain_label	oval-sc:EntityItemStringType (0..1)	An selinux domain label associated with the process.
session_id	oval-sc:EntityItemIntType (0..1)	The session ID of the process.

< routingtable_item >

The routingtable_item holds information about an individual routing table entry found in a system's primary routing table. Each routingtable_item contains a destination IP address, gateway, netmask, flags, and the name of the interface associated with it. It is important to note that only numerical addresses will be collected and that their symbolic representations will not be resolved. This equivalent to using the '-n' option with route(8) or netstat(8). It extends the standard ItemType as defined in the oval-system-characteristics schema and one should refer to the ItemType description for more information.

Extends: oval-sc:ItemType

Child Elements

Table 978: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
destination	oval-sc:EntityItemIPAddressType (0..1)	The destination IP address prefix of the routing table entry. This is the destination IP address and netmask/prefix-length expressed using CIDR notation.
gateway	oval-sc:EntityItemIPAddressType (0..1)	The gateway of the specified routing table entry.
flags	unix-sc:EntityItemRoutingTableFlagsType (0..unbounded)	The flags associated with the specified routing table entry.
interface_name	oval-sc:EntityItemStringType (0..1)	The name of the interface associated with the routing table entry.

< runlevel_item >

The runlevel item holds information about the start or kill state of a specified service at a given runlevel. Each runlevel item contains service name and runlevel information as well as start and kill information. It extends the standard ItemType as defined in the oval-system-characteristics schema and one should refer to the ItemType description for more information.

Extends: oval-sc:ItemType

Child Elements

Table 979: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
service_name	oval-sc:EntityItemStringType (0..1)	The service_name entity is the actual name of the specific service.
runlevel	oval-sc:EntityItemStringType (0..1)	The runlevel entity specifies the system runlevel associated with a service.
start	oval-sc:EntityItemBoolType (0..1)	The start entity specifies whether the service is scheduled to start at the runlevel.
kill	oval-sc:EntityItemBoolType (0..1)	The kill entity specifies whether the service is scheduled to be killed at the runlevel.

< sccs_item > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.10
- Reason: The sccs_item has been deprecated because the Source Code Control System (SCCS) is obsolete. The sccs_item may be removed in a future version of the language.

Extends: oval-sc:ItemType

Child Elements

Table 980: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-sc:EntityItemStringType (0..1)	Specifies the absolute path to an SCCS file. A directory cannot be specified as a filepath.
path	oval-sc:EntityItemStringType (0..1)	The path element specifies the directory component of the absolute path to an SCCS file.
filename	oval-sc:EntityItemStringType (0..1)	The name of an SCCS file.
module_name	oval-sc:EntityItemStringType (0..1)	
module_type	oval-sc:EntityItemStringType (0..1)	
release	oval-sc:EntityItemStringType (0..1)	
level	oval-sc:EntityItemStringType (0..1)	
branch	oval-sc:EntityItemStringType (0..1)	
sequence	oval-sc:EntityItemStringType (0..1)	
what_string	oval-sc:EntityItemStringType (0..1)	

< shadow_item >

/etc/shadow. See shadow(4).

Extends: oval-sc:ItemType

Child Elements

Table 981: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
username	oval-sc:EntityItemStringType (0..1)	This is the name of the user for which data was gathered.
password	oval-sc:EntityItemStringType (0..1)	This is the encrypted version of the user's password.
chg_lst	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	This is the date of the last password change in days since 1/1/1970.
chg_allow	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	This specifies how often in days a user may change their password. It can also be thought of as the minimum age of a password.
chg_req	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	This describes how long the user can keep a password before the system forces them to change it.
exp_warn	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	This describes how long before password expiration the system begins warning the user. The system will warn the user at each login.
exp_inact	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	This describes how many days of account inactivity the system will wait after a password expires before locking the account? This window, usually only set to a few days, gives users who are logging in very seldomly a bit of extra time to receive the password expiration warning and change their password.
exp_date	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	This specifies when will the account's password expire, in days since 1/1/1970.
flag	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	This is a numeric reserved field that the shadow file may use in the future.
encrypt_method	unix-sc:EntityItemEncryptMethodType (0..1)	The encrypt_method entity describes method that is used for hashing passwords.

< symlink_item >

The symlink_item element identifies the result generated for a symlink_object.

Extends: oval-sc:ItemType

Child Elements

Table 982: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-sc:EntityItemStringType (1..1)	Specifies the filepath to the subject symbolic link file, specified by the symlink_object.
canonical_path	oval-sc:EntityItemStringType (1..1)	Specifies the canonical path for the target of the symbolic link file specified by the filepath.

< sysctl_item >

The sysctl_item stores information retrieved from the local system about a kernel parameter and its respective value(s).

Extends: oval-sc:ItemType

Child Elements

Table 983: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	The name element contains a string that represents the name of a kernel parameter that was collected from the local system.
value	oval-sc:EntityItemAnySimpleType (0..unbounded)	The value element contains a string that represents the current value(s) for the specified kernel parameter on the local system.

< uname_item >

Information about the hardware the machine is running on. This information is the parsed equivalent of uname -a.

Extends: oval-sc:ItemType

Child Elements

Table 984: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
machine_class	oval-sc:EntityItemStringType (0..1)	This entity specifies the machine hardware name. This corresponds to the command uname -m.
node_name	oval-sc:EntityItemStringType (0..1)	This entity specifies the host name. This corresponds to the command uname -n.
os_name	oval-sc:EntityItemStringType (0..1)	This entity specifies the operating system name. This corresponds to the command uname -s.
os_release	oval-sc:EntityItemStringType (0..1)	This entity specifies the build version. This corresponds to the command uname -r.
os_version	oval-sc:EntityItemStringType (0..1)	This entity specifies the operating system version. This corresponds to the command uname -v.
processor_type	oval-sc:EntityItemStringType (0..1)	This entity specifies the processor type. This corresponds to the command uname -p.

< xinetd_item >

The xinetd item holds information associated with different Internet services. It extends the standard ItemType as defined in the oval-system-characteristics schema and one should refer to the ItemType description for more information.

Extends: oval-sc:ItemType

Child Elements

Table 985: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
protocol	oval-sc:EntityItemStringType (0..1)	The protocol entity specifies the protocol that is used by the service. The list of valid protocols can be found in /etc/protocols.
service_name	oval-sc:EntityItemStringType (0..1)	The service_name entity specifies the name of the service.
flags	oval-sc:EntityItemStringType (0..unbounded)	The flags entity specifies miscellaneous settings associated with the service.
no_access	oval-sc:EntityItemStringType (0..unbounded)	The no_access entity specifies the remote hosts to which the service is unavailable. Please see the xinetd.conf(5) man page for information on the different formats that can be used to describe a host.
only_from	oval-sc:EntityItemIPAddressStringType (0..unbounded)	The only_from entity specifies the remote hosts to which the service is available. Please see the xinetd.conf(5) man page for information on the different formats that can be used to describe a host.
port	oval-sc:EntityItemIntType (0..1)	The port entity specifies the port used by the service.
server	oval-sc:EntityItemStringType (0..1)	The server entity specifies the executable that is used to launch the service.
server_arguments	oval-sc:EntityItemStringType (0..1)	The server_arguments entity specifies the arguments that are passed to the executable when launching the service.
socket_type	oval-sc:EntityItemStringType (0..1)	The socket_type entity specifies the type of socket that is used by the service. Possible values include: stream, dgram, raw, or seqpacket.
type	unix-sc:EntityItemXinetdTypeStatusType (0..unbounded)	The type entity specifies the type of the service. A service may have multiple types.
user	oval-sc:EntityItemStringType (0..1)	The user entity specifies the user identifier of the process that is running the service. The user identifier may be expressed as a numerical value or as a user name that exists in /etc/passwd.
wait	oval-sc:EntityItemBoolType (0..1)	The wait entity specifies whether or not the service is single-threaded or multi-threaded and whether or not xinetd accepts the connection or the service accepts the connection. A value of 'true' indicates that the service is single-threaded and the service will accept the connection. A value of 'false' indicates that the service is multi-threaded and xinetd will accept the connection.
disabled	oval-sc:EntityItemBoolType (0..1)	The disabled entity specifies whether or not the service is disabled. A value of 'true' indicates that the service is disabled and will not start. A value of 'false' indicates that the service is not disabled.

== EntityItemCapabilityType ==

The EntityItemCapabilityType complex type restricts a string value to a specific set of values that describe POSIX capability types associated with a process service. This list is based off the values defined in linux/include/linux/capability.h. Documentation on each allowed value can be found in capability.h. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 986: Enumeration Values

Value	Description
CAP_CHOWN	
CAP_DAC_OVERRIDE	
CAP_DAC_READ_SEARCH	
CAP_FOWNER	
CAP_FSETID	
CAP_KILL	
CAP_SETGID	
CAP_SETUID	
CAP_SETPCAP	
CAP_LINUX_IMMUTABLE	
CAP_NET_BIND_SERVICE	
CAP_NET_BROADCAST	
CAP_NET_ADMIN	

Continued on next page

Table 986 – continued from previous page

Value	Description
CAP_NET_RAW	
CAP_IPC_LOCK	
CAP_IPC_OWNER	
CAP_SYS_MODULE	
CAP_SYS_RAWIO	
CAP_SYS_CHROOT	
CAP_SYS_PTRACE	
CAP_SYS_ADMIN	
CAP_SYS_BOOT	
CAP_SYS_NICE	
CAP_SYS_RESOURCE	
CAP_SYS_TIME	
CAP_SYS_TTY_CONFIG	
CAP_MKNOD	
CAP_LEASE	
CAP_AUDIT_WRITE	

Continued on next page

Table 986 – continued from previous page

Value	Description
CAP_AUDIT_CONTROL	
CAP_SETFCAP	
CAP_MAC_OVERRIDE	
CAP_MAC_ADMIN	
CAP_SYS_PACCT	
CAP_SYSLOG	
CAP_WAKE_ALARM	
CAP_BLOCK_SUSPEND	
CAP_AUDIT_READ	
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityItemEndpointType ==

The EntityItemEndpointType complex type restricts a string value to a specific set of values that describe endpoint types associated with an Internet service. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 987: Enumeration Values

Value	Description
stream	The stream value is used to describe a stream socket.
dgram	The dgram value is used to describe a datagram socket.
raw	The raw value is used to describe a raw socket.
seqpacket	The seqpacket value is used to describe a sequenced packet socket.
tli	The tli value is used to describe all TLI endpoints.
sunrpc_tcp	The sunrpc_tcp value is used to describe all SUNRPC TCP endpoints.
sunrpc_udp	The sunrpc_udp value is used to describe all SUNRPC UDP endpoints.
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemGconfTypeType ==

The EntityItemGconfTypeType complex type restricts a string value to the seven values GCONF_VALUE_STRING, GCONF_VALUE_INT, GCONF_VALUE_FLOAT, GCONF_VALUE_BOOL, GCONF_VALUE_SCHEMA, GCONF_VALUE_LIST, and GCONF_VALUE_PAIR that specify the type of the value associated with a GConf preference key. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 988: Enumeration Values

Value	Description
GCONF_VALUE_STRING	The GCONF_VALUE_STRING type is used to describe a preference key that has a string value.
GCONF_VALUE_INT	The GCONF_VALUE_INT type is used to describe a preference key that has a integer value.
GCONF_VALUE_FLOAT	The GCONF_VALUE_FLOAT type is used to describe a preference key that has a float value.
GCONF_VALUE_BOOL	The GCONF_VALUE_BOOL type is used to describe a preference key that has a boolean value.
GCONF_VALUE_SCHEMA	The GCONF_VALUE_SCHEMA type is used to describe a preference key that has a schema value. The actual value will be the default value as specified in the GConf schema.
GCONF_VALUE_LIST	The GCONF_VALUE_LIST type is used to describe a preference key that has a list of values. The actual values will be one of the primitive GConf datatypes GCONF_VALUE_STRING, GCONF_VALUE_INT, GCONF_VALUE_FLOAT, GCONF_VALUE_BOOL, and GCONF_VALUE_SCHEMA. Note that all of the values associated with a GCONF_VALUE_LIST are required to have the same type.
GCONF_VALUE_PAIR	The GCONF_VALUE_PAIR type is used to describe a preference key that has a pair of values. The actual values will consist of the primitive GConf datatypes GCONF_VALUE_STRING, GCONF_VALUE_INT, GCONF_VALUE_FLOAT, GCONF_VALUE_BOOL, and GCONF_VALUE_SCHEMA. Note that the values associated with a GCONF_VALUE_PAIR are not required to have the same type.
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemRoutingTableFlagsType ==

The EntityItemRoutingTableFlagsType complex type restricts a string value to a specific set of values that describe the flags associated with a routing table entry. This list is based off the values defined in the man pages of various platforms. For Linux, please see route(8). For Solaris, please see netstat(1M). For HP-UX, please see netstat(1). For Mac OS, please see netstat(1). For FreeBSD, please see netstat(1). Documentation on each allowed value can be found in the previously listed man pages. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 989: Enumeration Values

Value	Description
UP	
GATEWAY	
HOST	
REINSTATE	
DYNAMIC	
MODIFIED	
ADDRCONF	
CACHE	
REJECT	
REDUNDANT	
SETSRC	
BROADCAST	
LOCAL	
PROTOCOL_1	
PROTOCOL_2	
PROTOCOL_3	
BLACK_HOLE	
CLONING	
PROTOCOL_CLONING	

The following table is a mapping between the generic flag enumeration values and the actual flag values found on the various platforms. If the flag value is not specified, for a particular generic flag enumeration value, the flag value is not defined for that platform. ` Name Linux Solaris HPUX Mac OS FreeBSD   AIX UP U U U U
U   U GATEWAY G G G G G   G HOST H H H H H   H REINSTATE
R DYNAMIC D D D D   D MODIFIED M M M   M ADDRCONF A A CACHE C
e REJECT ! R R   R REDUNDANT M (>=9) SETSRC S BROADCAST B b
b   b LOCAL L   l PROTOCOL_1 1 1   1 PROTOCOL_2 2
2   2 PROTOCOL_3 3 3   3 BLACK_HOLE B B CLONING C C   c
PROTOCOL_CLONING c c INTERFACE_SCOPE I LINK_LAYER L L   L MULTICAST
m   m STATIC S S   S WAS_CLONED W W   W XRESOLVE X
X USABLE   u PINNED
P ACTIVE_DEAD_GATEWAY_DETECTION   A
(>=5.1) `

== EntityItemXinetdTypeStatusType ==

The EntityItemXinetdTypeStatusType complex type restricts a string value to five values, either RPC, INTERNAL, UNLISTED, TCPMUX, or TCPMUXPLUS that specify the type of service registered in xinetd. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 990: Enumeration Values

Value	Description
INTERNAL	The INTERNAL type is used to describe services like echo, chargen, and others whose functionality is supplied by xinetd itself.
RPC	The RPC type is used to describe services that use remote procedure call ala NFS.
UNLISTED	The UNLISTED type is used to describe services that aren't listed in /etc/protocols or /etc/rpc.
TCPMUX	The TCPMUX type is used to describe services that conform to RFC 1078. This type indiciates that the service is responsible for handling the protocol handshake.
TCPMUXPLUS	The TCPMUXPLUS type is used to describe services that conform to RFC 1078. This type indicates that xinetd is responsible for handling the protocol handshake.
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemWaitStatusType ==

The EntityItemWaitStatusType complex type restricts a string value to two values, either wait or nowait, that specify whether the server that is invoked by inetd will take over the listening socket associated with the service, and whether once launched, inetd will wait for that server to exit, if ever, before it resumes listening for new service requests. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 991: Enumeration Values

Value	Description
wait	The value of 'wait' specifies that the server that is invoked by inetd will take over the listening socket associated with the service, and once launched, inetd will wait for that server to exit, if ever, before it resumes listening for new service requests.
nowait	The value of 'nowait' specifies that the server that is invoked by inetd will not wait for any existing server to finish before taking over the listening socket associated with the service.
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemEncryptMethodType ==

The EntityItemEncryptMethodType complex type restricts a string value to a set that corresponds to the allowed encrypt methods used for protected passwords in a shadow file. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 992: Enumeration Values

Value	Description
DES	The DES method corresponds to the (none) prefix.
BSDi	The BSDi method corresponds to BSDi modified DES or the ‘_’ prefix.
MD5	The MD5 method corresponds to MD5 for Linux/BSD or the 1 prefix.
Blowfish	The Blowfish method corresponds to Blowfish (OpenBSD) or the 2 or $2a$ prefixes.
Sun MD5	The Sun MD5 method corresponds to the $md5$ prefix.
SHA-256	The SHA-256 method corresponds to the 5 prefix.
SHA-512	The SHA-512 method corresponds to the 6 prefix.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityItemInterfaceType ==

The EntityItemInterfaceType complex type restricts a string value to a specific set of values. These values describe the different interface types which are defined in ‘if_arp.h’. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-sc:EntityItemStringType

Table 993: Enumeration Values

Value	Description
ARPHRD_ETHER	The ARPHRD_ETHER type is used to describe ethernet interfaces.
ARPHRD_FDDI	The ARPHRD_FDDI type is used to describe fiber distributed data interfaces (FDDI).
ARPHRD_LOOPBACK	The ARPHRD_LOOPBACK type is used to describe loopback interfaces.
ARPHRD_VOID	The ARPHRD_VOID type is used to describe unknown interfaces.
ARPHRD_PPP	The ARPHRD_PPP type is used to describe point-to-point protocol interfaces (PPP).
ARPHRD_SLIP	The ARPHRD_SLIP type is used to describe serial line internet protocol interfaces (SLIP).
ARPHRD_PRONET	The ARPHRD_PRONET type is used to describe PROnet token ring interfaces.
	The empty string value is permitted here to allow for detailed error reporting.

Open Vulnerability and Assessment Language: MacOS Definition

- Schema: MacOS Definition
- Version: 5.11.1:1.2
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the MacOS specific tests found in Open Vulnerability and Assessment Language (OVAL). Each test is an extension of the standard test element defined in the Core Definition Schema. Through extension, each test inherits a set of elements and attributes that are shared amongst all OVAL tests. Each test is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core Definition Schema is not outlined here.

The MacOS Definition Schema was initially developed by The Center for Internet Security. Many thanks to their contributions to OVAL and the security community.

The OVAL Schema is maintained by the 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.cisecurity.org.

Test Listing

- *< accountinfo_test >*
- *< authorizationdb_test >*
- *< corestorage_test >*
- *< diskutil_test >*
- *< gatekeeper_test >*
- *< inetlisteningservers_test > (Deprecated)* (Deprecated)
- *< inetlisteningserver510_test >*
- *< keychain_test >*
- *< launchd_test >*
- *< nvram_test >*
- *< plist_test > (Deprecated)* (Deprecated)
- *< plist510_test > (Deprecated)* (Deprecated)
- *< plist511_test >*
- *< pwpolicy_test > (Deprecated)* (Deprecated)
- *< pwpolicy59_test >*
- *< rlimit_test >*
- *< softwareupdate_test >*
- *< systemprofiler_test >*
- *< systemsetup_test >*

< accountinfo_test >

User account information (username, uid, gid, etc.) See netinfo(5) for field information, niutil(1) for retrieving it. As of Mac OS 10.5, niutil(1) is no longer available, however, the same functionality can be obtained using dscl(1). Specifically, the command ‘dscl . -list /Users’ can be used to list all users and the command ‘dscl . -read /Users/some_user passwd uid gid realname home shell’ can be used to retrieve the attributes associated with an account.

Extends: oval-def:TestType

Child Elements

Table 994: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< accountinfo_object >

The accountinfo_object element is used by an accountinfo_test to define the object(s) to be evaluated. This object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An accountinfo_object consists of a single username that identifies the account from which to gather information.

Extends: oval-def:ObjectType

Child Elements

Table 995: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
username	oval-def:EntityObjectStringType (1..1)	Specifies the user of the account to gather information from.
oval-def:filter	n/a (0..unbounded)	

< accountinfo_state >

The accountinfo_state element defines the different information that can be used to evaluate the specified accounts. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 996: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
username	oval-def:EntityStateStringType (0..1)	Specifies the user of the account to gather information from.
password	oval-def:EntityStateStringType (0..1)	Obfuscated (*) or encrypted password for this user.
uid	oval-def:EntityStateIntType (0..1)	The numeric user id, or uid, is the third column of each user's entry in /etc/passwd. This element represents the owner of the file.
gid	oval-def:EntityStateIntType (0..1)	Group ID of this account.
realname	oval-def:EntityStateStringType (0..1)	User's real name, aka gecos field of /etc/passwd.
home_dir	oval-def:EntityStateStringType (0..1)	The home directory for this user account.
login_shell	oval-def:EntityStateStringType (0..1)	The login shell for this user account.

< authorizationdb_test >

The authorizationdb_test is used to check the properties of the plist-style XML output from the "security authorizationdb read >right-name<" command, for reading information about rights authorizations on MacOSX. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an authorizationdb_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 997: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< authorizationdb_object >

The authorizationdb_object element is used by an authorizationdb_test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An authorizationdb_object consists of a right_name entity that contains the name of the right to be read from the authorization dabatase. The resulting plist data can be queried using the xpath entity.

Extends: oval-def:ObjectType

Child Elements

Table 998: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
right_name	oval-def:EntityObjectStringType (1..1)	Specifies the right name to be queried (read) from the authorization database.
xpath	oval-def:EntityObjectStringType (1..1)	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 given xpath would be impossible.
oval-def:filter	n/a (0..unbounded)	

< authorizationdb_state >

The authorizationdb_state element defines a value used to evaluate the result of a specific authorizationdb_object item.

Extends: oval-def:StateType

Child Elements

Table 999: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
right_name	oval-def:EntityStateStringType (0..1)	Specifies the right_name used to create the object.
xpath	oval-def:EntityStateStringType (0..1)	Specifies an Xpath expression describing the text node(s) or attribute(s) to look at.
value_of	oval-def:EntityStateAnySimpleType (0..1)	The value_of element checks the value(s) of the text node(s) or attribute(s) found.

< corestorage_test >

The corestorage_test is used to check the properties of the plist-style XML output from the “diskutil cs list -plist” command, for reading information about the CoreStorage setup on MacOSX. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an corestorage_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 1000: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< corestorage_object >

The corestorage_object element is used by an corestorage_test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An corestorage_object consists of a uuid entity that contains the UUID of the volume whose information should be read (i.e., ‘diskutil cs info -plist [UUID]’). The resulting plist data can be queried using the xpath entity.

Extends: oval-def:ObjectType

Child Elements

Table 1001: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
uuid	oval-def:EntityObjectStringType (1..1)	Specifies the UUID of the volume about which the plist information should be retrieved.
xpath	oval-def:EntityObjectStringType (1..1)	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 given xpath would be impossible.
oval-def:filter	n/a (0..unbounded)	

< corestorage_state >

The corestorage_state element defines a value used to evaluate the result of a specific corestorage_object item.

Extends: oval-def:StateType

Child Elements

Table 1002: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
uuid	oval-def:EntityStateStringType (0..1)	Specifies the UUID of the volume about which the plist information was retrieved.
xpath	oval-def:EntityStateStringType (0..1)	Specifies an Xpath expression describing the text node(s) or attribute(s) to look at.
value_of	oval-def:EntityStateAnySimpleType (0..1)	The value_of element checks the value(s) of the text node(s) or attribute(s) found.

< diskutil_test >

The diskutil_test is used to verify packages on a Mac OS system. The information used by this test is modeled after the diskutil command's verifyPermissions option. On MacOS X 10.11 and later, this option was replaced by the repair_packages command. For more information, see diskutil(8) or repair_packages(8). It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a diskutil_object and the optional diskutil_state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 1003: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< diskutil_object >

The diskutil_object element is used by a diskutil_test to define the volumes containing packages to be verified on a Mac OS system. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 1004: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
device	oval-def:EntityObjectStringType (1..1)	The device entity is a string that represents the name of a volume containing system packages that is mounted on a Mac OS system to verify. Please see diskutil(8) or repair_packages(8) for instructions on how to specify the volume.
filepath	oval-def:EntityObjectStringType (1..1)	The filepath element specifies the absolute path for a file or directory in the specified package
oval-def:filter	n/a (0..unbounded)	

< diskutil_state >

The diskutil_state element defines the different verification information associated with a disk on a Mac OS system. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1005: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
device	oval-def:EntityStateStringType (0..1)	The device entity is a string that represents the volume on a Mac OS system to verify. Please see diskutil(8) or repair_packages(8) for instructions on how to specify the device.
filepath	oval-def:EntityStateStringType (0..1)	The filepath element specifies the absolute path for a file or directory on the specified device.
uread	macos-def:EntityStatePermissionCompareType (0..1)	Has the actual user read permission changed from the expected user read permission?
uwrite	macos-def:EntityStatePermissionCompareType (0..1)	Has the actual user write permission changed from the expected user write permission?
uexec	macos-def:EntityStatePermissionCompareType (0..1)	Has the actual user exec permission changed from the expected user exec permission?
gread	macos-def:EntityStatePermissionCompareType (0..1)	Has the actual group read permission changed from the expected group read permission?
gwrite	macos-def:EntityStatePermissionCompareType (0..1)	Has the actual group write permission changed from the expected group write permission?
gexec	macos-def:EntityStatePermissionCompareType (0..1)	Has the actual group exec permission changed from the expected group exec permission?
oread	macos-def:EntityStatePermissionCompareType (0..1)	Has the actual others read permission changed from the expected others read permission?
owrite	macos-def:EntityStatePermissionCompareType (0..1)	Has the actual others write permission changed from the expected others write permission?
oexec	macos-def:EntityStatePermissionCompareType (0..1)	Has the actual others exec permission changed from the expected others exec permission?
user_differs	oval-def:EntityStateBoolType (0..1)	Has the actual user changed from the expected user?
actual_user	oval-def:EntityStateIntType (0..1)	The actual user of the file/directory.
expected_user	oval-def:EntityStateIntType (0..1)	The expected user of the file/directory.
group_differs	oval-def:EntityStateBoolType (0..1)	Has the actual group changed from the expected group?
actual_group	oval-def:EntityStateIntType (0..1)	The actual group of the file/directory.
expected_group	oval-def:EntityStateIntType (0..1)	The expected group of the file/directory.
symlink_diff	oval-def:EntityStateBoolT	Has the actual symlink changed from the expected symlink?

< gatekeeper_test >

The gatekeeper_test is used to check the status of Gatekeeper and any unsigned applications that have been granted execute permission.

Extends: oval-def:TestType

Child Elements

Table 1006: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< gatekeeper_object >

The gatekeeper_object is a singleton used to access information about Gatekeeper.

Extends: oval-def:ObjectType

< gatekeeper_state >

The gatekeeper_state element makes it possible to make assertions about Gatekeeper's operational status and unsigned applications that have been granted execute permission.

Extends: oval-def:StateType

Child Elements

Table 1007: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
enabled	oval-def:EntityStateBoolType (0..1)	The status of Gatekeeper assessments.
unlabeled	oval-def:EntityStateStringType (0..1)	The path to an unsigned application folder to which Gatekeeper has granted execute permission.

< inetlisteningservers_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.10
- Reason: The inetlisteningservers_test has been deprecated and replaced by the inetlisteningserver510_test. The name of an application cannot be used to uniquely identify an application that is listening on the network. As a result, the inetlisteningserver510_object utilizes the protocol, local_address, and local_port entities to uniquely identify an application listening on the network. Please see the inetlisteningserver510_test for additional information.

This test's purpose is generally used to check if an application is listening on the network, either for a new connection or as part of an ongoing connection. This is limited to applications that are listening for connections that use the TCP or UDP protocols and have addresses represented as IPv4 or IPv6 addresses (AF_INET or AF_INET6). It is generally speaking the parsed output of running the command netstat -tuwlnpe with root privilege.

Extends: oval-def:TestType

Child Elements

Table 1008: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< inetlisteningservers_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.10
- Reason: The inetlisteningservers_object has been deprecated and replaced by the inetlisteningserver510_object. The name of an application cannot be used to uniquely identify an application that is listening on the network. As a result, the inetlisteningserver510_object utilizes the protocol, local_address, and local_port entities to uniquely identify an application listening on the network. Please see the inetlisteningserver510_object for additional information.

The inetlisteningservers_object element is used by an inetlisteningserver test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 1009: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
program_name	oval-def:EntityObjectStringType (1..1)	
oval-def:filter	n/a (0..unbounded)	

< inetlisteningservers_state > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.10
- Reason: The inetlisteningservers_state has been deprecated and replaced by the inetlisteningserver510_state. The name of an application cannot be used to uniquely identify an application that is listening on the network. As a result, the inetlisteningserver510_object utilizes the protocol, local_address, and local_port entities to uniquely identify an application listening on the network. Please see the inetlisteningserver510_state for additional information.

The inetlisteningservers_state element defines the different information that can be used to evaluate the specified inet listening server. This includes the local address, foreign address, port information, and process id. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1010: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
program_name	oval-def:EntityStateStringType (0..1)	This is the name of the communicating program.
local_address	oval-def:EntityStateIPAddressStringType (0..1)	This is the IP address of the network interface on which the program listens. Note that the IP address can be IPv4 or IPv6.
local_full_address	oval-def:EntityStateStringType (0..1)	This is the IP address and network port on which the program listens, equivalent to local_address:local_port. Note that the IP address can be IPv4 or IPv6.
local_port	oval-def:EntityStateIntType (0..1)	This is the TCP or UDP port on which the program listens. Note that this is not a list – if a program listens on multiple ports, or on a combination of TCP and UDP, each will have its own entry in the table data stored by this test.
foreign_address	oval-def:EntityStateIPAddressStringType (0..1)	This is the IP address with which the program is communicating, or with which it will communicate, in the case of a listening server. Note that the IP address can be IPv4 or IPv6.
foreign_full_address	oval-def:EntityStateStringType (0..1)	This is the IP address and network port to which the program is communicating or will accept communications from, equivalent to foreign_address:foreign_port. Note that the IP address can be IPv4 or IPv6.
foreign_port	oval-def:EntityStateStringType (0..1)	This is the TCP or UDP port to which the program communicates. In the case of a listening program accepting new connections, this is usually ‘0’.
pid	oval-def:EntityStateIntType (0..1)	This is the process ID of the process. The process in question is that of the program communicating on the network.
protocol	oval-def:EntityStateStringType (0..1)	This is the transport-layer protocol, in lowercase: tcp or udp.
user_id	oval-def:EntityStateStringType (0..1)	The numeric user id, or uid, is the third column of each user’s entry in /etc/passwd. It represents the owner, and thus privilege level, of the specified program.

< inetlisteningserver510_test >

The inetlisteningserver510_test is used to check if an application is listening on the network, either for a new connection or as part of an ongoing connection. This is limited to applications that are listening for connections that use the TCP or UDP protocols and have addresses represented as IPv4 or IPv6 addresses (AF_INET or AF_INET6). One method for retrieving the required information is by parsing the output of the command ‘lsof -i -P -n -l’ with root privileges.

Extends: oval-def:TestType

Child Elements

Table 1011: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< inetlisteningserver510_object >

The inetlisteningserver510_object element is used by an inetlisteningserver510_test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 1012: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
protocol	oval-def:EntityObjectStringType (1..1)	The protocol entity defines a certain transport-layer protocol, in lowercase: tcp or udp.
local_address	oval-def:EntityObjectIPAddressStringType (1..1)	This is the IP address of the network interface on which an application listens. Note that the IP address can be IPv4 or IPv6.
local_port	oval-def:EntityObjectIntType (1..1)	This is the TCP or UDP port on which an application would listen. Note that this is not a list – if a program listens on multiple ports, or on a combination of TCP and UDP, each will be represented by its own object.
oval-def:filter	n/a (0..unbounded)	

< inetlisteningserver510_state >

The inetlisteningserver510_state element defines the different information that can be used to evaluate the specified inet listening server. This includes the local address, foreign address, port information, and process id. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1013: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
protocol	oval-def:EntityStateStringType (0..1)	This is the transport-layer protocol, in lowercase: tcp or udp.
local_address	oval-def:EntityStateIPAddressStringType (0..1)	This is the IP address of the network interface on which the program listens. Note that the IP address can be IPv4 or IPv6.
local_port	oval-def:EntityStateIntType (0..1)	This is the TCP or UDP port on which the program listens. Note that this is not a list – if a program listens on multiple ports, or on a combination of TCP and UDP, each will have its own entry in the table data stored by this test.
local_full_address	oval-def:EntityStateStringType (0..1)	This is the IP address and network port on which the program listens, equivalent to local_address:local_port. Note that the IP address can be IPv4 or IPv6.
program_name	oval-def:EntityStateStringType (0..1)	This is the name of the communicating program.
foreign_address	oval-def:EntityStateIPAddressStringType (0..1)	This is the IP address with which the program is communicating, or with which it will communicate, in the case of a listening server. Note that the IP address can be IPv4 or IPv6.
foreign_port	oval-def:EntityStateIntType (0..1)	This is the TCP or UDP port to which the program communicates. In the case of a listening program accepting new connections, this is usually '0'.
foreign_full_address	oval-def:EntityStateStringType (0..1)	This is the IP address and network port to which the program is communicating or will accept communications from, equivalent to foreign_address:foreign_port. Note that the IP address can be IPv4 or IPv6.
pid	oval-def:EntityStateIntType (0..1)	This is the process ID of the process. The process in question is that of the program communicating on the network.
user_id	oval-def:EntityStateIntType (0..1)	The numeric user id, or uid, is the third column of each user's entry in /etc/passwd. It represents the owner, and thus privilege level, of the specified program.

< keychain_test >

The keychain_test is used to check the properties of the plist-style XML output from the "security show-keychain-info >keychain<" command, for reading information about keychain settings on MacOSX. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an keychain_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 1014: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< keychain_object >

The keychain_object element is used by an corestorage_test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A keychain_object consists of a keychain (name) entity that contains the name of the keychain whose settings will be queried.

Extends: oval-def:ObjectType

Child Elements

Table 1015: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-def:EntityObjectStringType (1..1)	Specifies the filepath of the keychain to be queried. The default keychain for a user is normally located at ~/Library/Keychains/login.keychain.
oval-def:filter	n/a (0..unbounded)	

< keychain_state >

The keychain_state element defines a value used to evaluate the result of a specific keychain_object item.

Extends: oval-def:StateType

Child Elements

Table 1016: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-def:EntityStateStringType (0..1)	Specifies the filepath of the keychain used to create the object.
lock_on_sleep	oval-def:EntityStateBoolType (0..1)	Specifies whether the keychain is configured to lock when the computer sleeps.
timeout	oval-def:EntityStateIntType (0..1)	Specifies the inactivity timeout (in seconds) for the keychain, or 0 if there is no timeout.

< launchd_test >

The launchd_test is used to check the status of daemons/agents loaded via the launchd service. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a launchd_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 1017: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< launchd_object >

The launchd_object element is used by a launchd_test to define the daemon/agent to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A launchd_object consists of a label (name) entity that contains the name of the agent/daemon whose attributes will be queried.

Extends: oval-def:ObjectType

Child Elements

Table 1018: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
label	oval-def:EntityObjectStringType (1..1)	Specifies the deamon to be queried.
oval-def:filter	n/a (0..unbounded)	

< launchd_state >

The launchd_state element defines a value used to evaluate the result of a specific launchd_object item.

Extends: oval-def:StateType

Child Elements

Table 1019: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
label	oval-def:EntityStateStringType (0..1)	Specifies the name of the agent/daemon used to create the object.
pid	oval-def:EntityStateIntType (0..1)	Specifies the process ID of the daemon (if any).
status	oval-def:EntityStateIntType (0..1)	Specifies the last exit code of the daemon (if any), or if $lt; 0, indicates the negative of the signal that interrupted processing. For example, a value of -15 would indicate that the job was terminated via a SIGTERM.

< nvram_test >

This test pulls data from the ‘nvram -p’ output.

Extends: oval-def:TestType

Child Elements

Table 1020: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< nvram_object >

The nvram_object element is used by a nvram test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 1021: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
nvram_var	oval-def:EntityObjectStringType (1..1)	
oval-def:filter	n/a (0..unbounded)	

< nvram_state >

This test pulls data from the ‘nvram -p’ output.

Extends: oval-def:StateType

Child Elements

Table 1022: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
nvram_var	oval-def:EntityStateStringType (0..1)	This specifies the nvram variable to check.
nvram_value	oval-def:EntityStateStringType (0..1)	This is the value of the associated nvram variable.

< plist_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.10
- Reason: Replaced by the plist510_test. This test references the plist_object which does not contain an instance entity. As a result, it is not possible to differentiate between two preference keys that have the same name using the plist_object. The plist510_test was added to address this deficiency. See the plist510_test.
- Comment: This test has been deprecated and may be removed in a future version of the language.

The plist_test is used to check the value(s) associated with property list preference keys. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a plist_object and the optional plist_state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 1023: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< plist_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.10
- Reason: Replaced by the plist510_object. This object does not contain an instance entity. As a result, it is not possible to differentiate between two preference keys that have the same name using this object. The plist510_object was added to address this deficiency. See the plist510_object.
- Comment: This object has been deprecated and may be removed in a future version of the language.

The plist_object element is used by a plist_test to define the preference keys to collect and where to look for them. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 1024: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
key	oval-def:EntityObjectStringType (1..1)	The preference key to check. If the xsi:nil attribute is set to 'true', the plist does not have any keys associated with it (i.e. it is not a CFDictionary) and the default value of the plist will be collected.
app_id	oval-def:EntityObjectStringType (1..1)	The unique application identifier that specifies the application to use when looking up the preference key (e.g. com.apple.Safari).
filepath	oval-def:EntityObjectStringType (1..1)	The absolute path to a plist file (e.g. ~/Library/Preferences/com.apple.Safari.plist). A directory cannot be specified as a filepath.
oval-def:filter	n/a (0..unbounded)	

< plist_state > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.10

- Reason: Replaced by the plist510_state. This state is used in conjunction with the plist_object which does not contain an instance entity. As a result, it is not possible to differentiate between two preference keys that have the same name using the plist_object. The plist510_state was added to address this deficiency. See the plist510_state.
- Comment: This object has been deprecated and may be removed in a future version of the language.

The plist_state element defines the different information that can be used to evaluate the specified property list preference key. This includes the preference key, application identifier, filepath, type, as well as the preference key's value. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1025: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
key	oval-def:EntityStateStringType (0..1)	The preference key to check.
app_id	oval-def:EntityStateStringType (0..1)	The unique application identifier that specifies the application to use when looking up the preference key (e.g. com.apple.Safari).
filepath	oval-def:EntityStateStringType (0..1)	The absolute path to a plist file (e.g. ~/Library/Preferences/com.apple.Safari.plist).
instance	oval-def:EntityStateIntType (0..1)	The instance of the preference key found in the plist. The first instance of a matching preference key is given the instance value of 1, the second instance of a matching preference key is given the instance value of 2, and so on. Note that the main purpose of this entity is to provide uniqueness for the different plist_items that result from multiple instances of a given preference key in the same plist file.
type	macos-def:EntityStatePlistTypeType (0..1)	The type of the preference key.
value	oval-def:EntityStateAnySimpleType (0..1)	The value of the preference key.

< plist510_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.11.2:1.0
- Reason: Replaced by the plist511_test. This test references the plist_object which cannot express the context hierarchy required to differentiate between nodes with identical names. As a result, it is not possible to address a particular node when the order of their parent nodes is indeterminate. The plist511_test was added to address this deficiency. See the plist511_test.

- Comment: This test has been deprecated and may be removed in a future version of the language.

The plist510_test is used to check the value(s) associated with property list preference keys. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a plist510_object and the optional plist510_state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 1026: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< plist510_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.11.2:1.0
- Reason: Replaced by the plist511_object. This object cannot express the context hierarchy required to differentiate between nodes with identical names. As a result, it is not possible to address a particular node when the order of their parent nodes is indeterminate. The plist511_object was added to address this deficiency. See the plist511_object.
- Comment: This object has been deprecated and may be removed in a future version of the language.

The plist510_object element is used by a plist510_test to define the preference keys to collect and where to look for them. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 1027: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
key	oval-def:EntityObjectStringType (1..1)	The preference key to check. If the xsi:nil attribute is set to 'true', the plist does not have any keys associated with it (i.e. it is not a CFDictionary) and the default value of the plist will be collected.
app_id	oval-def:EntityObjectStringType (1..1)	The unique application identifier that specifies the application to use when looking up the preference key (e.g. com.apple.Safari).
filepath	oval-def:EntityObjectStringType (1..1)	The absolute path to a plist file (e.g. ~/Library/Preferences/com.apple.Safari.plist). A directory cannot be specified as a filepath.
instance	oval-def:EntityObjectIntType (1..1)	The instance of the preference key found in the plist. The first instance of a matching preference key is given the instance value of 1, the second instance of a matching preference key is given the instance value of 2, and so on. Instance values must be assigned using a depth-first approach. Note that the main purpose of this entity is to provide uniqueness for the different plist_items that result from multiple instances of a given preference key in the same plist file.
oval-def:filter	n/a (0..unbounded)	

< plist510_state > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.11.2:1.0
- Reason: Replaced by the plist511_state. This state is used in conjunction with the plist510_object which cannot express the context hierarchy required to differentiate between nodes with identical names. As a result, it is not possible to address a particular node when the order of their parent nodes is indeterminate. The plist511_state was added to address this deficiency. See the plist511_state.
- Comment: This object has been deprecated and may be removed in a future version of the language.

The plist510_state element defines the different information that can be used to evaluate the specified property list preference key. This includes the preference key, application identifier, filepath, type, as well as the preference key's value. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1028: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
key	oval-def:EntityStateStringType (0..1)	The preference key to check.
app_id	oval-def:EntityStateStringType (0..1)	The unique application identifier that specifies the application to use when looking up the preference key (e.g. com.apple.Safari).
filepath	oval-def:EntityStateStringType (0..1)	The absolute path to a plist file (e.g. ~/Library/Preferences/com.apple.Safari.plist).
instance	oval-def:EntityStateIntType (0..1)	The instance of the preference key found in the plist. The first instance of a matching preference key is given the instance value of 1, the second instance of a matching preference key is given the instance value of 2, and so on. Instance values must be assigned using a depth-first approach. Note that the main purpose of this entity is to provide uniqueness for the different plist_items that result from multiple instances of a given preference key in the same plist file.
type	macos-def:EntityStatePlistTypeType (0..1)	The type of the preference key.
value	oval-def:EntityStateAnySimpleType (0..1)	The value of the preference key.

< plist511_test >

The plist511_test is used to check the value(s) associated with property list preference keys. It can be used to represent any plist file in XML form (whether its native format is ASCII text, binary, or XML), permitting the use of the XPATH query language to explore its contents. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a plist511_object and the optional plist511_state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 1029: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< plist511_object >

The plist511_object element is used by a plist511_test to define the preference keys to collect and where to look for them. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 1030: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
app_id	oval-def:EntityObjectStringType (1..1)	The unique application identifier that specifies the application to use when looking up the preference key (e.g. com.apple.Safari).
filepath	oval-def:EntityObjectStringType (1..1)	The absolute path to a plist file (e.g. /Library/Preferences/com.apple.TimeMachine.plist). A directory cannot be specified as a filepath.
xpath	oval-def:EntityObjectStringType (1..1)	Specifies an XPath 1.0 expression to evaluate against the XML representation of the plist file specified by the filename or app_id entity. This XPath 1.0 expression must evaluate to a list of zero or more text values which will be accessible in OVAL via instances of the value_of item entity. Any results from evaluating the XPath 1.0 expression other than a list of text strings (e.g., a nodes set) is considered an error. The intention is that the text values be drawn from instances of a single, uniquely named element or attribute. However, an OVAL interpreter is not required to verify this, so the author should define the XPath expression carefully. Note that “equals” is the only valid operator for the xpath entity.
oval-def:filter	n/a (0..unbounded)	

< plist511_state >

The plist511_state element defines the different information that can be used to evaluate the specified property list preference key. This includes the preference key, application identifier, filepath, type, as well as the preference key’s value. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1031: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
app_id	oval-def:EntityStateStringType (0..1)	The unique application identifier that specifies the application to use when looking up the preference key (e.g. com.apple.Safari).
filepath	oval-def:EntityStateStringType (0..1)	The absolute path to a plist file (e.g. ~/Library/Preferences/com.apple.Safari.plist).
xpath	oval-def:EntityStateStringType (0..1)	Specifies an XPath expression describing the text node(s) or attribute(s) to look at.
value_of	oval-def:EntityStateAnySimpleType (0..1)	The value of the preference key.

< pwpolicy_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.9
- Reason: Replaced by the pwpolicy59_test. The username, userpass, and directory_node entities in the pwpolicy_object, pwpolicy_state, and pwpolicy_item were underspecified and as a result their meaning was uncertain. A new test was created to resolve this issue. See the pwpolicy59_test.
- Comment: This test has been deprecated and may be removed in a future version of the language.

This test pulls data from the 'pwpolicy -getpolicy' output. The actual values get stored under /var/db/netinfo/local.nidb/ in a Store.# file. Is this test actually needed, or can the text file content test be used instead?

Extends: oval-def:TestType

Child Elements

Table 1032: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< pwpolicy_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.9

- Reason: Replaced by the pwpolicy59_object. The username, userpass, and directory_node entities in the pwpolicy_object were underspecified and as a result their meaning was uncertain. A new object was created to resolve this issue. See the pwpolicy59_object.
- Comment: This object has been deprecated and may be removed in a future version of the language.

The pwpolicy_object element is used by a pwpolicy_test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 1033: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
username	oval-def:EntityObjectStringType (1..1)	
userpass	oval-def:EntityObjectStringType (1..1)	
directory_node	oval-def:EntityObjectStringType (1..1)	
oval-def:filter	n/a (0..unbounded)	

< pwpolicy_state > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.9
- Reason: Replaced by the pwpolicy59_state. The username, userpass, and directory_node entities in the pwpolicy_state were underspecified and as a result their meaning was uncertain. A new state was created to resolve this issue. See the pwpolicy59_state.
- Comment: This state has been deprecated and may be removed in a future version of the language.

Extends: oval-def:StateType

Child Elements

Table 1034: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
username	oval-def:EntityStateStringType (0..1)	
userpass	oval-def:EntityStateStringType (0..1)	
directory_node	oval-def:EntityStateStringType (0..1)	
maxChars	oval-def:EntityStateIntType (0..1)	Maximum number of characters allowed in a password.
maxFailedLoginAttempts	oval-def:EntityStateIntType (0..1)	Maximum number of failed logins before the account is locked.
minChars	oval-def:EntityStateIntType (0..1)	Minimum number of characters allowed in a password.
passwordCannotBeName	oval-def:EntityStateBoolType (0..1)	Defines if the password is allowed to be the same as the username or not.
requiresAlpha	oval-def:EntityStateBoolType (0..1)	Defines if the password must contain an alphabetical character or not.
requiresNumeric	oval-def:EntityStateBoolType (0..1)	Defines if the password must contain an numeric character or not.

< pwpolicy59_test >

This test retrieves password policy data from the ‘pwpolicy -getpolicy -u target_user [-a username] [-p userpass] [-n directory_node]’ output where username, userpass, and directory_node are optional. Please see the ‘pwpolicy’ man page for additional information.

Extends: oval-def:TestType

Child Elements

Table 1035: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< pwpolicy59_object >

The pwpolicy59_object element is used by a pwpolicy59_test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 1036: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
target_user	oval-def:EntityObjectStringType (1..1)	The target_user element specifies the user whose password policy information should be collected. If an operation other than equals is specified, the users on the system should be enumerated and the 'pwpolicy' command should be issued for each user that matches the target_user element. If the xsi:nil attribute is set to true, the global policy should be retrieved.
username	oval-def:EntityObjectStringType (1..1)	The username element specifies the username of the authenticator. If the xsi:nil attribute is set to true, authentication to the directory node will not be performed (i.e. the '-a' and '-p' command line options will not be specified when issuing the 'pwpolicy' command) and the xsi:nil attribute of the userpass element should also be set to true.
userpass	oval-def:EntityObjectStringType (1..1)	The userpass element specifies the password of the authenticator as specified by the username element. If the xsi:nil attribute is set to true, authentication to the directory node will not be performed (i.e. the '-a' and '-p' command line options will not be specified when issuing the 'pwpolicy' command) and the xsi:nil attribute of the username element should also be set to true.
directory_node	oval-def:EntityObjectStringType (1..1)	The directory_node element specifies the directory node that you would like to retrieve the password policy information from. If the xsi:nil attribute is set to true, the default directory node is used (i.e. the '-n' command line option will not be specified when issuing the 'pwpolicy' command).
oval-def:filter	n/a (0..unbounded)	

< pwpolicy59_state >

The pwpolicy59_state element defines the different information that can be used to evaluate the password policy for the target user in the specified directory node. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1037: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
target_user	oval-def:EntityStateStringType (0..1)	The target_user element specifies the user whose password policy information should be collected.
username	oval-def:EntityStateStringType (0..1)	The username element specifies the username of the authenticator.
userpass	oval-def:EntityStateStringType (0..1)	The userpass element specifies the password of the authenticator as specified by the username element.
directory_node	oval-def:EntityStateStringType (0..1)	The directory_node element specifies the directory node that you would like to retrieve the password policy information from.
maxChars	oval-def:EntityStateIntType (0..1)	Maximum number of characters allowed in a password.
maxFailedLoginAttempts	oval-def:EntityStateIntType (0..1)	Maximum number of failed logins before the account is locked.
minChars	oval-def:EntityStateIntType (0..1)	Minimum number of characters allowed in a password.
passwordCannotBeName	oval-def:EntityStateBoolType (0..1)	Defines if the password is allowed to be the same as the username or not.
requiresAlpha	oval-def:EntityStateBoolType (0..1)	Defines if the password must contain an alphabetical character or not.
requiresNumeric	oval-def:EntityStateBoolType (0..1)	Defines if the password must contain an numeric character or not.
maxMinutesUntilChangePassword	oval-def:EntityStateIntType (0..1)	Maximum number of minutes until the password must be changed.
minMinutesUntilChangePassword	oval-def:EntityStateIntType (0..1)	Minimum number of minutes between password changes.
requiresMixedCase	oval-def:EntityStateBoolType (0..1)	Defines if the password must contain upper and lower case characters or not.
requiresSymbol	oval-def:EntityStateBoolType (0..1)	Defines if the password must contain a symbol character or not.
minutesUntilFailedLoginReset	oval-def:EntityStateIntType (0..1)	Number of minutes after login has been disabled due to too many failed login attempts to wait before reenabling login.
usingHistory	oval-def:EntityStateIntType (0..1)	0 = user can reuse the current pass-word, 1 = user cannot reuse the current password, 2-15 = user cannot reuse the last n passwords.
canModifyPassword-	oval-def:EntityStateBoolType	If true, the user can change the password.

< rlimit_test >

The rlimit_test is used to check system resource limits for launchd. It is a singleton object. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The state element specifies the system setup elements to check.

Extends: oval-def:TestType

Child Elements

Table 1038: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< rlimit_object >

The rlimit_object is a singleton used to access resource limit information.

Extends: oval-def:ObjectType

< rlimit_state >

The rlimit_state element makes it possible to make assertions about the resource limits for launchd.

A resource limit is specified as a soft (current) limit and a hard (max) limit. When a soft limit is exceeded a process may receive a signal (for example, if the cpu time or file size is exceeded), but it will be allowed to con-tinue continue tinue execution until it reaches the hard limit (or modifies its resource limit).

For any ‘unlimited’ resource, the entity will have the status of ‘does not exist’.

Extends: oval-def:StateType

Child Elements

Table 1039: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
cpu_current	oval-def:EntityStateIntType (0..1)	The maximum amount of cpu time (in seconds) to be used by each process.
cpu_max	oval-def:EntityStateIntType (0..1)	cpu hard limit.
filesize_current	oval-def:EntityStateIntType (0..1)	The largest size (in bytes) file that may be created.
filesize_max	oval-def:EntityStateIntType (0..1)	filesize hard limit.
data_current	oval-def:EntityStateIntType (0..1)	The maximum size (in bytes) of the data segment for a process; this defines how far a program may extend its break with the sbrk(2) system call.
data_max	oval-def:EntityStateIntType (0..1)	data hard limit.
stack_current	oval-def:EntityStateIntType (0..1)	The maximum size (in bytes) of the stack segment for a process; this defines how far a program's stack segment may be extended. Stack extension is performed automatically by the system.
stack_max	oval-def:EntityStateIntType (0..1)	stack hard limit.
core_current	oval-def:EntityStateIntType (0..1)	The largest size (in bytes) core file that may be created.
core_max	oval-def:EntityStateIntType (0..1)	core hard limit.
rss_current	oval-def:EntityStateIntType (0..1)	The maximum size (in bytes) to which a process's resident set size may grow. This imposes a limit on the amount of physical memory to be given to a process; if memory is tight, the system will prefer to take memory from processes that are exceeding their declared resident set size.
rss_max	oval-def:EntityStateIntType (0..1)	rss hard limit.
memlock_current	oval-def:EntityStateIntType (0..1)	The maximum size (in bytes) which a process may lock into memory using the mlock(2) function.
memlock_max	oval-def:EntityStateIntType (0..1)	memlock hard limit.
maxproc_current	oval-def:EntityStateIntType (0..1)	The maximum number of simultaneous processes for this user id.
maxproc_max	oval-def:EntityStateIntType (0..1)	maxproc hard limit.
maxfiles_current	oval-def:EntityStateIntType (0..1)	The maximum number of open files for this process.

< softwareupdate_test >

The softwareupdate_test is used to check the status of automatic software updates on MacOSX. It is a singleton object. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The state element specifies the softwareupdate elements to check.

Extends: oval-def:TestType

Child Elements

Table 1040: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< softwareupdate_object >

The softwareupdate_object is a singleton used to access automatic software update information.

Extends: oval-def:ObjectType

< softwareupdate_state >

The softwareupdate_state element makes it possible to make assertions about the state of automatic software updates.

Extends: oval-def:StateType

Child Elements

Table 1041: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
schedule	oval-def:EntityStateBoolType (0..1)	Specifies whether automatic checking is enabled (true).
software_title	oval-def:EntityStateStringType (0..1)	Specifies the title string for an available (not installed) software update.

< systemprofiler_test >

The systemprofiler_test is used to check the properties of the plist-style XML output from the "system_profiler -xml <data type>" command, for reading information about system inventory data on MacOSX. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an systemprofiler_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 1042: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< systemprofiler_object >

The systemprofiler_object element is used by an systemprofiler_test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An systemprofiler_object consists of a data_type entity that contains the name of the datatype that was probed by the system_profiler utility. The resulting plist data can be queried using the xpath entity.

Extends: oval-def:ObjectType

Child Elements

Table 1043: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
data_type	macos-def:EntityObjectDataTypeType (1..1)	The data_type entity provides the datatype value that is desired.
xpath	oval-def:EntityObjectStringType (1..1)	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 given xpath would be impossible.
oval-def:filter	n/a (0..unbounded)	

< systemprofiler_state >

The systemprofiler_state element defines a value used to evaluate the result of a specific systemprofiler_object item.

Extends: oval-def:StateType

Child Elements

Table 1044: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
data_type	macos-def:EntityStateDataTypeType (0..1)	The data_type entity provides the datatype value that is desired.
xpath	oval-def:EntityStateStringType (0..1)	Specifies an Xpath expression describing the text node(s) or attribute(s) to look at.
value_of	oval-def:EntityStateAnySimpleType (0..1)	The value_of element checks the value(s) of the text node(s) or attribute(s) found.

< systemsetup_test >

The systemsetup_test is used to check systemsetup properties. It is a singleton object. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The state element specifies the system setup elements to check.

Extends: oval-def:TestType

Child Elements

Table 1045: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< systemsetup_object >

The systemsetup_object is a singleton used to access system setup information.

Extends: oval-def:ObjectType

< systemsetup_state >

The systemsetup_state element makes it possible to make assertions about system setup settings.

Extends: oval-def:StateType

Child Elements

Table 1046: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
timezone	oval-def:EntityStateStringType (0..1)	Specifies the name of the time zone.
usingnetworktime	oval-def:EntityStateBoolType (0..1)	Specifies weather the machine is using network time.
networktimeserver	oval-def:EntityStateStringType (0..1)	Specifies the network time server.
computersleep	oval-def:EntityStateIntType (0..1)	Specifies the computer sleep inactivity timer, or 0 for never.
displaysleep	oval-def:EntityStateIntType (0..1)	Specifies the display sleep inactivity timer, or 0 for never.
harddisksleep	oval-def:EntityStateIntType (0..1)	Specifies the hard disk sleep inactivity timer, or 0 for never.
wakeonmodem	oval-def:EntityStateBoolType (0..1)	Specifies whether the computer will wake up if the modem is accessed.
wakeonnetworkaccess	oval-def:EntityStateBoolType (0..1)	Specifies whether the computer will wake up if the network is accessed.
restartfreeze	oval-def:EntityStateBoolType (0..1)	Specifies whether the computer will restart after freezing.
allowpowerbuttontosleepcomputer	oval-def:EntityStateBoolType (0..1)	Specifies whether the power button can be used to cause the computer to sleep.
remotelogin	oval-def:EntityStateBoolType (0..1)	Specifies whether remote logins are allowed.
remoteappleevents	oval-def:EntityStateBoolType (0..1)	Specifies whether remote Apple events are enabled.
computername	oval-def:EntityStateStringType (0..1)	Specifies the computer's name.
startupdisk	oval-def:EntityStateStringType (0..1)	Specifies the startup disk.
waitforstartupafterpowerfailure	oval-def:EntityStateIntType (0..1)	Specifies the number of seconds the computer waits to start up after a power failure.
disablekeyboardwhenenclosurelockisengaged	oval-def:EntityStateBoolType (0..1)	Specifies whether the keyboard is locked when the closure lock is engaged.
kernelbootarchitecturesetting	oval-def:EntityStateStringType (0..1)	Specifies the kernel boot architecture setting.

== EntityObjectDataTypeType ==

The EntityObjectDataTypeType complex type defines the different values that are valid for the data_type entity of a system_profiler object. These values describe the system_profiler XML data to be retrieved. The empty string is also allowed as a valid value to support an empty element that is found when a variable reference is used within the index entity. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values. Please note that the values identified are for the data_type entity and are not valid values for the datatype attribute.

Restricts: oval-def:EntityObjectStringType

Table 1047: Enumeration Values

Value	Description
SPHardwareDataType	(No Description)
SPNetworkDataType	(No Description)
SPSoftwareDataType	(No Description)
SPParallelATADataType	(No Description)
SPAudioDataType	(No Description)
SPBluetoothDataType	(No Description)
SPDiagnosticsDataType	(No Description)
SPDiscBurningDataType	(No Description)
SPEthernetDataType	(No Description)
SPFibreChannelDataType	(No Description)
SPFireWireDataType	(No Description)
SPDisplaysDataType	(No Description)
SPHardwareRAIDDataType	(No Description)
SPMemoryDataType	(No Description)
SPPCIDataType	(No Description)
SPParallelSCSIDataType	(No Description)
SPPowerDataType	(No Description)
SPPrintersDataType	(No Description)
SPSASDataType	(No Description)
SPSerialATADataType	(No Description)
SPUSBDataType	(No Description)
SPAirPortDataType	(No Description)
SPFirewallDataType	(No Description)
SPNetworkLocationDataType	(No Description)
SPModemDataType	(No Description)
SPNetworkVolumeDataType	(No Description)
SPWWANDataType	(No Description)
SPApplicationsDataType	(No Description)
SPDeveloperToolsDataType	(No Description)
SPExtensionsDataType	(No Description)
SPFontsDataType	(No Description)
SPFrameworksDataType	(No Description)
SPLogsDataType	(No Description)
SPManagedClientDataType	(No Description)
SPPrefPaneDataType	(No Description)
SPStartupItemDataType	(No Description)
SPSyncServicesDataType	(No Description)
SPUniversalAccessDataType	(No Description)

Continued on next page

Table 1047 – continued from previous page

Value	Description
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateDataTypeType ==

The EntityStateDataTypeType complex type defines the different values that are valid for the data_type entity of a system_profiler state. These values describe the system_profiler XML data to be retrieved. The empty string is also allowed as a valid value to support an empty element that is found when a variable reference is used within the index entity. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values. Please note that the values identified are for the data_type entity and are not valid values for the datatype attribute.

Restricts: oval-def:EntityObjectStringType

Table 1048: Enumeration Values

Value	Description
SPHardwareDataType	(No Description)
SPNetworkDataType	(No Description)
SPSoftwareDataType	(No Description)
SPParallelATADataType	(No Description)
SPAudioDataType	(No Description)
SPBluetoothDataType	(No Description)
SPDiagnosticsDataType	(No Description)
SPDiscBurningDataType	(No Description)
SPEthernetDataType	(No Description)
SPFibreChannelDataType	(No Description)
SPFireWireDataType	(No Description)
SPDisplaysDataType	(No Description)
SPHardwareRAIDDataType	(No Description)
SPMemoryDataType	(No Description)
SPPCIDataType	(No Description)
SPParallelSCSIDataType	(No Description)
SPPowerDataType	(No Description)
SPPrintersDataType	(No Description)
SPSASDataType	(No Description)
SPSerialATADataType	(No Description)
SPUSBDataType	(No Description)
SPAirPortDataType	(No Description)
SPFirewallDataType	(No Description)
SPNetworkLocationDataType	(No Description)
SPModemDataType	(No Description)
SPNetworkVolumeDataType	(No Description)
SPWWANDataType	(No Description)
SPApplicationsDataType	(No Description)
SPDeveloperToolsDataType	(No Description)
SPExtensionsDataType	(No Description)
SPFontsDataType	(No Description)

Continued on next page

Table 1048 – continued from previous page

Value	Description
SPFrameworksDataType	(No Description)
SPLogsDataType	(No Description)
SPManagedClientDataType	(No Description)
SPPrefPaneDataType	(No Description)
SPStartupItemDataType	(No Description)
SPSyncServicesDataType	(No Description)
SPUniversalAccessDataType	(No Description)
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStatePermissionCompareType ==

The EntityStatePermissionCompareType complex type restricts a string value to more, less, or same which specifies if an actual permission is different than the expected permission (more or less restrictive) or if the permission is the same. The empty string is also allowed to support empty elements associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 1049: Enumeration Values

Value	Description
more	The actual permission is more restrictive than the expected permission.
less	The actual permission is less restrictive than the expected permission.
same	The actual permission is the same as the expected permission.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStatePlistTypeType == (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.11.2:1.0

- Reason: Used only by the deprecated plist_state and plist510_state.
- Comment: This enumeration has been deprecated and may be removed in a future version of the language.

The EntityStatePlistTypeType complex type restricts a string value to the seven values CFString, CFNumber, CFBoolean, CFDate, CFData, CFArray, and CFDictionary that specify the datatype of the value associated with a property list preference key. The empty string is also allowed to support empty elements associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 1050: Enumeration Values

Value	Description
CFString	The CFString type is used to describe a preference key that has a string value. The OVAL string datatype should be used to represent CFString values.
CFNumber	The CFNumber type is used to describe a preference key that has a integer or float value. The OVAL int and float datatypes should be used, as appropriate, to represent CFNumber values.
CFBoolean	The CFBoolean type is used to describe a preference key that has a boolean value. The OVAL boolean datatype should be used to represent CFBoolean values.
CFDate	The CFDate type is used to describe a preference key that has a date value. The OVAL string datatype should be used to represent CFDate values.
CFData	The CFData type is used to describe a preference that has a base64-encoded binary value. The OVAL string datatype should be used to represent CFData values.
CFArray	The CFArray type is used to describe a preference key that has a collection of values. This is represented as multiple value entities.
CFDictionary	The CFDictionary type is used to describe a preference key that has a collection of key-value pairs. Note that the collection of CFDictionary values is not supported. If an attempt is made to collect a CFDictionary value, an error should be reported.
	The empty string value is permitted here to allow for empty elements associated with variable references.

Open Vulnerability and Assessment Language: MacOS System Characteristics

- Schema: MacOS System Characteristics
- Version: 5.11.1:1.2
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the MacOS specific system characteristic items found in Open Vulnerability and Assessment Language (OVAL). Each item is an extension of the standard test element defined in the Core Definition Schema. Through extension, each test inherits a set of elements and attributes that are shared amongst all OVAL tests. Each test is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core Definition Schema is not outlined here.

The MacOS System Characteristics Schema was initially developed by The Center for Internet Security. Many thanks to their contributions to OVAL and the security community.

The OVAL Schema is maintained by the 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.cisecurity.org.

Item Listing

- *< accountinfo_item >*
- *< authorizationdb_item >*
- *< corestorage_item >*
- *< diskutil_item >*
- *< gatekeeper_item >*
- *< inetlisteningserver_item > (Deprecated)*
- *< inetlisteningserver510_item >*
- *< keychain_item >*
- *< launchd_item >*
- *< nvram_item >*
- *< plist_item > (Deprecated)*
- *< plist511_item >*
- *< pwpolicy_item > (Deprecated)*
- *< pwpolicy59_item >*
- *< rlimit_item >*
- *< softwareupdate_item >*
- *< systemprofiler_item >*
- *< systemsetup_item >*

< accountinfo_item >

This item stores sser account information (username, uid, gid, etc.).

Extends: oval-sc:ItemType

Child Elements

Table 1051: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
username	oval-sc:EntityItemStringType (0..1)	The user associated with the information collected.
password	oval-sc:EntityItemStringType (0..1)	Obfuscated (*) or encrypted password for this user.
uid	oval-sc:EntityItemIntType (0..1)	The numeric user id, or uid, is the third column of each user's entry in /etc/passwd. This element represents the owner of the file.
gid	oval-sc:EntityItemIntType (0..1)	Group ID of this account.
realname	oval-sc:EntityItemStringType (0..1)	User's real name, aka gecos field of /etc/passwd.
home_dir	oval-sc:EntityItemStringType (0..1)	The home directory for this user account.
login_shell	oval-sc:EntityItemStringType (0..1)	The login shell for this user account.

< authorizationdb_item >

This item stores results from checking the contents of an authorizationdb right.

Extends: oval-sc:ItemType

Child Elements

Table 1052: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
right_name	oval-sc:EntityItemStringType (0..1)	Specifies the right_name in which the item is specified.
xpath	oval-sc:EntityItemStringType (0..1)	Specifies an Xpath expression describing the text node(s) or attribute(s) to look at.
value_of	oval-sc:EntityItemAnySimpleType (0..unbounded)	The value_of element checks the value(s) of the text node(s) or attribute(s) found. How this is used is entirely controlled by operator attributes.

< corestorage_item >

This item stores results from checking the contents of the CoreStorage XML plist information.

Extends: oval-sc:ItemType

Child Elements

Table 1053: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
uuid	oval-sc:EntityItemStringType (1..1)	Specifies the UUID of the volume about which the plist information was retrieved.
xpath	oval-sc:EntityItemStringType (0..1)	Specifies an Xpath expression describing the text node(s) or attribute(s) to look at.
value_of	oval-sc:EntityItemAnySimpleType (0..unbounded)	The value_of element checks the value(s) of the text node(s) or attribute(s) found. How this is used is entirely controlled by operator attributes.

< diskutil_item >

The diskutil_item holds verification information about an individual disk on a Mac OS system. Each diskutil_item contains a device, filepath, and details on how the actual permissions, ownerships and link targets differ from the expected values. For more information, see diskutil(8) or repair_packages(8). It extends the standard ItemType as defined in the oval-system-characteristics schema and one should refer to the ItemType description for more information.

Extends: oval-sc:ItemType

Child Elements

Table 1054: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
device	oval-sc:EntityItemStringType (0..1)	The device entity is a string that represents the disk on a Mac OS system to verify. Please see diskutil(8) for instructions on how to specify the device.
filepath	oval-sc:EntityItemStringType (0..1)	The filepath element specifies the absolute path for a file or directory on the specified device.
uread	macos-sc:EntityItemPermissionCompareType (0..1)	Has the actual user read permission changed from the expected user read permission?
uwrite	macos-sc:EntityItemPermissionCompareType (0..1)	Has the actual user write permission changed from the expected user write permission?
uexec	macos-sc:EntityItemPermissionCompareType (0..1)	Has the actual user exec permission changed from the expected user exec permission?
gread	macos-sc:EntityItemPermissionCompareType (0..1)	Has the actual group read permission changed from the expected group read permission?
gwrite	macos-sc:EntityItemPermissionCompareType (0..1)	Has the actual group write permission changed from the expected group write permission?
gexec	macos-sc:EntityItemPermissionCompareType (0..1)	Has the actual group exec permission changed from the expected group exec permission?
oread	macos-sc:EntityItemPermissionCompareType (0..1)	Has the actual others read permission changed from the expected others read permission?
owrite	macos-sc:EntityItemPermissionCompareType (0..1)	Has the actual others write permission changed from the expected others write permission?
oexec	macos-sc:EntityItemPermissionCompareType (0..1)	Has the actual others exec permission changed from the expected others exec permission?
user_differs	oval-sc:EntityItemBoolType (0..1)	Has the actual user changed from the expected user?
actual_user	oval-sc:EntityItemIntType (0..1)	The actual user of the file/directory.
expected_user	oval-sc:EntityItemIntType (0..1)	The expected user of the file/directory.
group_differs	oval-sc:EntityItemBoolType (0..1)	Has the actual group changed from the expected group?
actual_group	oval-sc:EntityItemIntType (0..1)	The actual group of the file/directory.
expected_group	oval-sc:EntityItemIntType (0..1)	The expected group of the file/directory.
sym-	oval-	Has the actual symlink changed from the expected symlink?

< gatekeeper_item >

This item stores results from checking the settings of the Gatekeeper.

Extends: oval-sc:ItemType

Child Elements

Table 1055: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
enabled	oval-sc:EntityItemBoolType (1..1)	The status of Gatekeeper assessments.
unlabeled	oval-sc:EntityItemStringType (0..unbounded)	The path to an unsigned application folder to which Gatekeeper has granted execute permission.

< inetlisteningserver_item > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.10
- Reason: The inetlisteningserver_item has been deprecated and replaced by the inetlisteningserver510_item. The name of an application cannot be used to uniquely identify an application that is listening on the network. As a result, the inetlisteningserver510_object utilizes the protocol, local_address, and local_port entities to uniquely identify an application listening on the network. Please see the inetlisteningserver510_item for additional information.

An inet listening server item stores the results of checking for network servers currently active on a system.

Extends: oval-sc:ItemType

Child Elements

Table 1056: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
program_name	oval-sc:EntityItemStringType (0..1)	This is the name of the communicating program.
local_address	oval-sc:EntityItemIPAddressStringType (0..1)	This is the IP address of the network interface on which the program listens. Note that the IP address can be IPv4 or IPv6.
local_full_address	oval-sc:EntityItemStringType (0..1)	This is the IP address and network port on which the program listens, equivalent to local_address:local_port. Note that the IP address can be IPv4 or IPv6.
local_port	oval-sc:EntityItemIntType (0..1)	This is the TCP or UDP port on which the program listens. Note that this is not a list – if a program listens on multiple ports, or on a combination of TCP and UDP, each will have its own entry in the table data stored by this item.
foreign_address	oval-sc:EntityItemIPAddressStringType (0..1)	This is the IP address with which the program is communicating, or with which it will communicate, in the case of a listening server. Note that the IP address can be IPv4 or IPv6.
foreign_full_address	oval-sc:EntityItemStringType (0..1)	This is the IP address and network port to which the program is communicating or will accept communications from, equivalent to foreign_address:foreign_port. Note that the IP address can be IPv4 or IPv6.
foreign_port	oval-sc:EntityItemStringType (0..1)	This is the TCP or UDP port to which the program communicates. In the case of a listening program accepting new connections, this is usually '0'.
pid	oval-sc:EntityItemIntType (0..1)	This is the process ID of the process. The process in question is that of the program communicating on the network.
protocol	oval-sc:EntityItemStringType (0..1)	This is the transport-layer protocol, in lowercase: tcp or udp.
user_id	oval-sc:EntityItemStringType (0..1)	The numeric user id, or uid, is the third column of each user's entry in /etc/passwd. It represents the owner, and thus privilege level, of the specified program.

< inetlisteningserver510_item >

An inet listening server item stores the results of checking for network servers currently active on a system.

Extends: oval-sc:ItemType

Child Elements

Table 1057: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
protocol	oval-sc:EntityItemStringType (0..1)	This is the transport-layer protocol, in lowercase: tcp or udp.
local_address	oval-sc:EntityItemIPAddressStringType (0..1)	This is the IP address of the network interface on which the program listens. Note that the IP address can be IPv4 or IPv6.
local_port	oval-sc:EntityItemIntType (0..1)	This is the TCP or UDP port on which the program listens. Note that this is not a list – if a program listens on multiple ports, or on a combination of TCP and UDP, each will have its own entry in the table data stored by this item.
local_full_address	oval-sc:EntityItemStringType (0..1)	This is the IP address and network port on which the program listens, equivalent to local_address:local_port. Note that the IP address can be IPv4 or IPv6.
program_name	oval-sc:EntityItemStringType (0..1)	This is the name of the communicating program.
foreign_address	oval-sc:EntityItemIPAddressStringType (0..1)	This is the IP address with which the program is communicating, or with which it will communicate, in the case of a listening server. Note that the IP address can be IPv4 or IPv6.
foreign_port	oval-sc:EntityItemIntType (0..1)	This is the TCP or UDP port to which the program communicates. In the case of a listening program accepting new connections, this is usually '0'.
foreign_full_address	oval-sc:EntityItemStringType (0..1)	This is the IP address and network port to which the program is communicating or will accept communications from, equivalent to foreign_address:foreign_port. Note that the IP address can be IPv4 or IPv6.
pid	oval-sc:EntityItemIntType (0..1)	This is the process ID of the process. The process in question is that of the program communicating on the network.
user_id	oval-sc:EntityItemIntType (0..1)	The numeric user id, or uid, is the third column of each user's entry in /etc/passwd. It represents the owner, and thus privilege level, of the specified program.

< keychain_item >

This item stores results from checking the settings of a keychain.

Extends: oval-sc:ItemType

Child Elements

Table 1058: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-sc:EntityItemStringType (1..1)	Specifies the filepath of the keychain.
lock_on_sleep	oval-sc:EntityItemBoolType (0..1)	Specifies the whether the keychain is configured to lock on sleep.
timeout	oval-sc:EntityItemIntType (0..1)	The inactivity timeout (in seconds) for the keychain, or 0 if there is no timeout.

< launchd_item >

This item stores results from checking a launchd-controlled daemon/agent.

Extends: oval-sc:ItemType

Child Elements

Table 1059: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
label	oval-sc:EntityItemStringType (1..1)	Specifies the name of the agent/daemon.
pid	oval-sc:EntityItemIntType (0..1)	Specifies the process ID of the daemon (if any).
status	oval-sc:EntityItemIntType (0..1)	Specifies the last exit code of the daemon (if any), or if $lt; 0, indicates the negative of the signal that interrupted processing. For example, a value of -15 would indicate that the job was terminated via a SIGTERM.

< nvram_item >

Output of ‘nvram -p’

Extends: oval-sc:ItemType

Child Elements

Table 1060: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
nvram_var	oval-sc:EntityItemStringType (0..1)	A nvram variabl.
nvram_value	oval-sc:EntityItemStringType (0..1)	This is the value of the associated nvram variable.

< plist_item > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.11.2:1.0
- Reason: The plist_item has been deprecated and replaced by the plist511_item. The plist_item cannot express the context hierarchy required to differentiate between nodes with identical names. As a result, it is not possible to address a particular node when the order of their parent nodes is indeterminate. The plist511_item was added to address this deficiency. See the plist511_item.

The plist_item holds information about an individual property list preference key found on a system. Each plist_item contains a preference key, application identifier or filepath, type, as well as the preference key's value. It extends the standard ItemType as defined in the oval-system-characteristics schema and one should refer to the ItemType description for more information.

Extends: oval-sc:ItemType

Child Elements

Table 1061: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
key	oval-sc:EntityItemStringType (0..1)	The preference key to check.
app_id	oval-sc:EntityItemStringType (0..1)	The unique application identifier that specifies the application to use when looking up the preference key (e.g. com.apple.Safari).
filepath	oval-sc:EntityItemStringType (0..1)	The absolute path to a plist file (e.g. ~/Library/Preferences/com.apple.Safari.plist).
instance	oval-sc:EntityItemIntType (0..1)	The instance of the preference key found in the plist. The first instance of a matching preference key is given the instance value of 1, the second instance of a matching preference key is given the instance value of 2, and so on. Instance values must be assigned using a depth-first approach. Note that the main purpose of this entity is to provide uniqueness for the different plist_items that result from multiple instances of a given preference key in the same plist file.
type	macos-sc:EntityItemPlistTypeType (0..1)	The type of the preference key.
value	oval-sc:EntityItemAnySimpleType (0..unbounded)	The value of the preference key.

< plist511_item >

The plist511_item stores results from checking the contents of the XML representation of a plist file. It extends the standard ItemType as defined in the oval-system-characteristics schema and one should refer to the ItemType description for more information.

Extends: oval-sc:ItemType

Child Elements

Table 1062: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
app_id	oval-sc:EntityItemStringType (0..1)	The unique application identifier that specifies the application to use when looking up the preferences key (e.g. com.apple.Safari).
filepath	oval-sc:EntityItemStringType (0..1)	The absolute path to a plist file (e.g. /Library/Preferences/com.apple.TimeMachine.plist).
xpath	oval-sc:EntityItemStringType (0..1)	Specifies an XPath 1.0 expression to evaluate against the XML representation of the plist file specified by the filename or app_id entity. This XPath 1.0 expression must evaluate to a list of zero or more text values which will be accessible in OVAL via instances of the value_of entity. Any results from evaluating the XPath 1.0 expression other than a list of text strings (e.g., a nodes set) is considered an error. The intention is that the text values be drawn from instances of a single, uniquely named element or attribute. However, an OVAL interpreter is not required to verify this, so the author should define the XPath expression carefully. Note that "equals" is the only valid operator for the xpath entity.
value_of	oval-sc:EntityItemAnySimpleType (0..unbounded)	The value_of element checks the value(s) of the text node(s) or attribute(s) found. How this is used is entirely controlled by operator attributes.

< pwpolicy_item > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.9
- Reason: Replaced by the pwpolicy59_item. The username, userpass, and directory_node entities in the pwpolicy_item were underspecified and as a result their meaning was uncertain. A new item was created to resolve this issue. See the pwpolicy59_item.
- Comment: This item has been deprecated and may be removed in a future version of the language.

Output of 'pwpolicy -getpolicy'. Please see the 'pwpolicy' man page for additional information.

Extends: oval-sc:ItemType

Child Elements

Table 1063: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
username	oval-sc:EntityItemStringType (0..1)	
userpass	oval-sc:EntityItemStringType (0..1)	
directory_node	oval-sc:EntityItemStringType (0..1)	
maxChars	oval-sc:EntityItemIntType (0..1)	Maximum number of characters allowed in a password.
maxFailedLoginAttempts	oval-sc:EntityItemIntType (0..1)	Maximum number of failed logins before the account is locked.
minChars	oval-sc:EntityItemIntType (0..1)	Minimum number of characters allowed in a password.
passwordCannotBeName	oval-sc:EntityItemBoolType (0..1)	Defines if the password is allowed to be the same as the username or not.
requiresAlpha	oval-sc:EntityItemBoolType (0..1)	Defines if the password must contain an alphabetical character or not.
requiresNumeric	oval-sc:EntityItemBoolType (0..1)	Defines if the password must contain an numeric character or not.

< pwpolicy59_item >

The pwpolicy59_item holds the password policy information for a particular user specified by the target_user element. Please see the ‘pwpolicy’ man page for additional information.

Extends: oval-sc:ItemType

Child Elements

Table 1064: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
target_user	oval-sc:EntityItemStringType (0..1)	The target_user element specifies the user whose password policy information was collected. If xsi:nil="true", the item specifies the global policy.
username	oval-sc:EntityItemStringType (0..1)	The username element specifies the username of the authenticator.
userpass	oval-sc:EntityItemStringType (0..1)	The userpass element specifies the password of the authenticator as specified by the username element.
directory_node	oval-sc:EntityItemStringType (0..1)	The directory_node element specifies the directory node that the password policy information was collected from.
maxChars	oval-sc:EntityItemIntType (0..1)	Maximum number of characters allowed in a password.
maxFailedLoginAttempts	oval-sc:EntityItemIntType (0..1)	Maximum number of failed logins before the account is locked.
minChars	oval-sc:EntityItemIntType (0..1)	Minimum number of characters allowed in a password.
passwordCannotBeName	oval-sc:EntityItemBoolType (0..1)	Defines if the password is allowed to be the same as the username or not.
requiresAlpha	oval-sc:EntityItemBoolType (0..1)	Defines if the password must contain an alphabetical character or not.
requiresNumeric	oval-sc:EntityItemBoolType (0..1)	Defines if the password must contain an numeric character or not.
maxMinutesUntilChangePassword	oval-sc:EntityItemIntType (0..1)	Maximum number of minutes until the password must be changed.
minMinutesUntilChangePassword	oval-sc:EntityItemIntType (0..1)	Minimum number of minutes between password changes.
requiresMixedCase	oval-sc:EntityItemBoolType (0..1)	Defines if the password must contain upper and lower case characters or not.
requiresSymbol	oval-sc:EntityItemBoolType (0..1)	Defines if the password must contain a symbol character or not.
minutesUntilFailedLoginReset	oval-sc:EntityItemIntType (0..1)	Number of minutes after login has been disabled due to too many failed login attempts to wait before reenabling login.
usingHistory	oval-sc:EntityItemIntType (0..1)	0 = user can reuse the current pass-word, 1 = user cannot reuse the current password, 2-15 = user cannot reuse the last n passwords.
canModifyPasswordforSelf	oval-sc:EntityItemBoolType (0..1)	If true, the user can change the password.

< rlimit_item >

The rlimit_item contains information about the resource limits for launchd.

A resource limit is specified as a soft (current) limit and a hard (max) limit. When a soft limit is exceeded a process may receive a signal (for example, if the cpu time or file size is exceeded), but it will be allowed to con-tinue continue tinue execution until it reaches the hard limit (or modifies its resource limit).

For any 'unlimited' resource, the entity will have the status of 'does not exist'.

Extends: oval-sc:ItemType

Child Elements

Table 1065: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
cpu_current	oval-sc:EntityItemIntType (1..1)	The maximum amount of cpu time (in seconds) to be used by each process.
cpu_max	oval-sc:EntityItemIntType (1..1)	cpu hard limit.
filesize_current	oval-sc:EntityItemIntType (1..1)	The largest size (in bytes) file that may be created.
filesize_max	oval-sc:EntityItemIntType (1..1)	filesize hard limit.
data_current	oval-sc:EntityItemIntType (1..1)	The maximum size (in bytes) of the data segment for a process; this defines how far a program may extend its break with the sbrk(2) system call.
data_max	oval-sc:EntityItemIntType (1..1)	data hard limit.
stack_current	oval-sc:EntityItemIntType (1..1)	The maximum size (in bytes) of the stack segment for a process; this defines how far a program's stack segment may be extended. Stack extension is performed automatically by the system.
stack_max	oval-sc:EntityItemIntType (1..1)	stack hard limit.
core_current	oval-sc:EntityItemIntType (1..1)	The largest size (in bytes) core file that may be created.
core_max	oval-sc:EntityItemIntType (1..1)	core hard limit.
rss_current	oval-sc:EntityItemIntType (1..1)	The maximum size (in bytes) to which a process's resident set size may grow. This imposes a limit on the amount of physical memory to be given to a process; if memory is tight, the system will prefer to take memory from processes that are exceeding their declared resident set size.
rss_max	oval-sc:EntityItemIntType (1..1)	rss hard limit.
memlock_current	oval-sc:EntityItemIntType (1..1)	The maximum size (in bytes) which a process may lock into memory using the mlock(2) function.
memlock_max	oval-sc:EntityItemIntType (1..1)	memlock hard limit.
maxproc_current	oval-sc:EntityItemIntType (1..1)	The maximum number of simultaneous processes for this user id.
maxproc_max	oval-sc:EntityItemIntType (1..1)	maxproc hard limit.
maxfiles_current	oval-sc:EntityItemIntType (1..1)	The maximum number of open files for this process.
	oval-	maxfiles hard limit.

< softwareupdate_item >

This item represents automatic software update information.

Extends: oval-sc:ItemType

Child Elements

Table 1066: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
schedule	oval-sc:EntityItemBoolType (1..1)	Specifies whether automatic checking is enabled (true).
software_title	oval-sc:EntityItemStringType (0..unbounded)	Specifies the title string for an available (not installed) software update.

< systemprofiler_item >

This item stores results from performing an XPATH query on the XML result of a systemprofiler data type query.

Extends: oval-sc:ItemType

Child Elements

Table 1067: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
data_type	macos-sc:EntityItemDataTypeType (0..1)	Specifies the data type that was used in collection.
xpath	oval-sc:EntityItemStringType (0..1)	Specifies an Xpath expression describing the text node(s) or attribute(s) to look at.
value_of	oval-sc:EntityItemAnySimpleType (0..unbounded)	The value_of element checks the value(s) of the text node(s) or attribute(s) found. How this is used is entirely controlled by operator attributes.

< systemsetup_item >

This item represents system setup information.

Extends: oval-sc:ItemType

Child Elements

Table 1068: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
timezone	oval-sc:EntityItemStringType (1..1)	Specifies the name of the current time zone.
usingnetworktime	oval-sc:EntityItemBoolType (1..1)	Specifies wither the machine is using network time.
networktimeserver	oval-sc:EntityItemStringType (0..1)	Specifies the network time server.
computersleep	oval-sc:EntityItemIntType (1..1)	Specifies the computer sleep inactivity timer, or 0 for never.
displaysleep	oval-sc:EntityItemIntType (1..1)	Specifies the display sleep inactivity timer, or 0 for never.
harddisksleep	oval-sc:EntityItemIntType (1..1)	Specifies the hard disk sleep inactivity timer, or 0 for never.
wakeonmodem	oval-sc:EntityItemBoolType (1..1)	Specifies whether the computer will wake up if the modem is accessed.
wakeonnetworkaccess	oval-sc:EntityItemBoolType (1..1)	Specifies whether the computer will wake up if the network is accessed.
restartfreeze	oval-sc:EntityItemBoolType (1..1)	Specifies whether the computer will restart after freezing.
restartpowerfailure	oval-sc:EntityItemBoolType (1..1)	Specifies whether the computer will restart after a power failure.
allowpowerbuttontosleepcomputer	oval-sc:EntityItemBoolType (1..1)	Specifies whether the power button can be used to cause the computer to sleep.
remotelogin	oval-sc:EntityItemBoolType (1..1)	Specifies whether remote logins are allowed.
remoteappleevents	oval-sc:EntityItemBoolType (0..1)	Specifies whether remote Apple events are enabled.
computername	oval-sc:EntityItemStringType (1..1)	Specifies the computer's name.
localsubnetname	oval-sc:EntityItemStringType (1..1)	Specifies the name of the local subnet.
startupdisk	oval-sc:EntityItemStringType (1..1)	Specifies the startup disks.
waitforstartupafterpowerfailure	oval-sc:EntityItemIntType (1..1)	Specifies the number of seconds the computer waits to start up after a power failure.
disablekeyboardwhenenclosurelockisengaged	oval-sc:EntityItemBoolType (1..1)	Specifies whether the keyboard is locked when the closure lock is engaged.

== EntityItemDataTypeType ==

The EntityItemDataTypeType complex type defines the different values that are valid for the data_type entity of a system_profiler item. These values describe the system_profiler XML data to be retrieved. The empty string is also allowed as a valid value to support an empty element that is found when a variable reference is used within the index entity. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values. Please note that the values identified are for the data_type entity and are not valid values for the datatype attribute.

Restricts: oval-sc:EntityItemStringType

Table 1069: Enumeration Values

Value	Description
SPHardwareDataType	(No Description)
SPNetworkDataType	(No Description)
SPSoftwareDataType	(No Description)
SPParallelATADataType	(No Description)
SPAudioDataType	(No Description)
SPBluetoothDataType	(No Description)
SPDiagnosticsDataType	(No Description)
SPDiscBurningDataType	(No Description)
SPEthernetDataType	(No Description)
SPFibreChannelDataType	(No Description)
SPFireWireDataType	(No Description)
SPDisplaysDataType	(No Description)
SPHardwareRAIDDataType	(No Description)
SPMemoryDataType	(No Description)
SPPCIDataType	(No Description)
SPParallelSCSIDataType	(No Description)
SPPowerDataType	(No Description)
SPPrintersDataType	(No Description)
SPSASDataType	(No Description)
SPSerialATADataType	(No Description)
SPUSBDataType	(No Description)
SPAirPortDataType	(No Description)
SPFirewallDataType	(No Description)
SPNetworkLocationDataType	(No Description)
SPModemDataType	(No Description)
SPNetworkVolumeDataType	(No Description)
SPWWANDataType	(No Description)
SPApplicationsDataType	(No Description)
SPDeveloperToolsDataType	(No Description)
SPExtensionsDataType	(No Description)
SPFontsDataType	(No Description)
SPFrameworksDataType	(No Description)
SPLogsDataType	(No Description)
SPManagedClientDataType	(No Description)
SPPrefPaneDataType	(No Description)
SPStartupItemDataType	(No Description)
SPSyncServicesDataType	(No Description)
SPUniversalAccessDataType	(No Description)

Continued on next page

Table 1069 – continued from previous page

Value	Description
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemPermissionCompareType ==

The EntityItemPermissionCompareType complex type restricts a string value to more, less, or same which specifies if an actual permission is different than the expected permission (more or less restrictive) or if the permission is the same. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 1070: Enumeration Values

Value	Description
more	The actual permission is more restrictive than the expected permission.
less	The actual permission is less restrictive than the expected permission.
same	The actual permission is the same as the expected permission.
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemPlistTypeType == (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.11.2:1.0
- Reason: Used only by the deprecated plist_item.
- Comment: This enumeration has been deprecated and may be removed in a future version of the language.

The EntityItemPlistTypeType complex type restricts a string value to the seven values CFString, CFNumber, CFBoolean, CFDate, CFData, CFArray, and CFDictionary that specify the type of the value associated with a property list preference key. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 1071: Enumeration Values

Value	Description
CFString	The CFString type is used to describe a preference key that has a string value. The OVAL string datatype should be used to represent CFString values.
CFNumber	The CFNumber type is used to describe a preference key that has a integer or float value. The OVAL int and float datatypes should be used, as appropriate, to represent CFNumber values.
CFBoolean	The CFBoolean type is used to describe a preference key that has a boolean value. The OVAL boolean datatype should be used to represent CFBoolean values.
CFDate	The CFDate type is used to describe a preference key that has a date value. The OVAL string datatype should be used to represent CFDate values.
CFData	The CFData type is used to describe a preference key that has a base64-encoded binary value. The OVAL string datatype should be used to represent CFData values.
CFArray	The CFArray type is used to describe a preference key that has a collection of values. This is represented as multiple value entities.
CFDictionary	The CFDictionary type is used to describe a preference key that has a collection of key-value pairs. Note that the collection of CFDictionary values is not supported. If an attempt is made to collect a CFDictionary value, an error should be reported.
	The empty string value is permitted here to allow for detailed error reporting.

Open Vulnerability and Assessment Language: FreeBSD Definition

- Schema: FreeBSD Definition
- Version: 5.11.1:1.1
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the FreeBSD specific tests found in Open Vulnerability and Assessment Language (OVAL). Each test is an extension of the standard test element defined in the Core Definition Schema. Through extension, each test inherits a set of elements and attributes that are shared amongst all OVAL tests. Each test is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core Definition Schema is not outlined here.

The OVAL Schema is maintained by the 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.cisecurity.org.

Test Listing

- *< portinfo_test >*

< portinfo_test >

The port info test is used to check the properties of a component of a FreeBSD system. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an portinfo_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 1072: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< portinfo_object >

The portinfo_object element is used by a port info test to define the specific FreeBSD package to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A port info object consists of a single pkginst element that identifies a specific package.

Extends: oval-def:ObjectType

Child Elements

Table 1073: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
pkginst	oval-def:EntityObjectStringType (1..1)	
oval-def:filter	n/a (0..unbounded)	

< portinfo_state >

The portinfo_state element defines the different information that can be used to evaluate the specified package. This includes the name, category, version, vendor, and description. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1074: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
pkginst	oval-def:EntityStateStringType (0..1)	
name	oval-def:EntityStateStringType (0..1)	The name of a package.
category	oval-def:EntityStateStringType (0..1)	
version	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	The version of a package.
vendor	oval-def:EntityStateStringType (0..1)	
description	oval-def:EntityStateStringType (0..1)	

Open Vulnerability and Assessment Language: FreeBSD System Characteristics

- Schema: FreeBSD System Characteristics
- Version: 5.11.1:1.1
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the FreeBSD specific system characteristic items found in Open Vulnerability and Assessment Language (OVAL). Each item is an extension of the standard test element defined in the Core Definition Schema. Through extension, each test inherits a set of elements and attributes that are shared amongst all OVAL tests. Each test is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core Definition Schema is not outlined here.

The OVAL Schema is maintained by the 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.cisecurity.org.

Item Listing

- *< portinfo_item >*

< portinfo_item >

Extends: oval-sc:ItemType

Child Elements

Table 1075: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
pkginst	oval-sc:EntityItemStringType (0..1)	
name	oval-sc:EntityItemStringType (0..1)	
category	oval-sc:EntityItemStringType (0..1)	
version	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	
vendor	oval-sc:EntityItemStringType (0..1)	
description	oval-sc:EntityItemStringType (0..1)	

Open Vulnerability and Assessment Language: HP-UX Definition

- Schema: HP-UX Definition
- Version: 5.11.1:1.1
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the HP-UX specific tests found in Open Vulnerability and Assessment Language (OVAL). Each test is an extension of the standard test element defined in the Core Definition Schema. Through extension, each test inherits a set of elements and attributes that are shared amongst all OVAL tests. Each test is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core Definition Schema is not outlined here.

The OVAL Schema is maintained by the 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.cisecurity.org.

Test Listing

- *< getconf_test >*
- *< ndd_test >*
- *< patch53_test >*
- *< patch_test > (Deprecated)* (Deprecated)
- *< swlist_test >*
- *< trusted_test >*

< getconf_test >

From /usr/bin/getconf. See getconf manpage for specific fields

Extends: oval-def:TestType

Child Elements

Table 1076: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< getconf_object >

Extends: oval-def:ObjectType

Child Elements

Table 1077: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
parameter_name	oval-def:EntityObjectStringType (1..1)	This is the parameter name to check.
pathname	oval-def:EntityObjectStringType (1..1)	This is the pathname to check. Note that pathname is optional in the getconf call. A nil pathname (empty wth attribute xsi:nil='true') in OVAL should be interpreted as if it was not supplied to the getconf call.
oval-def:filter	n/a (0..unbounded)	

< getconf_state >

Extends: oval-def:StateType

Child Elements

Table 1078: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
parameter_name	oval-def:EntityStateStringType (0..1)	This is the parameter name to check
pathname	oval-def:EntityStateStringType (0..1)	This is the pathname to check. Note that pathname is optional in the getconf call. A nil pathname in OVAL should be interpreted as if it was not supplied to the getconf call.
output	oval-def:EntityStateAnySimpleType (0..1)	The output produced by the getconf command.

< ndd_test >

From /usr/bin/ndd. See ndd manpage for specific fields

Extends: oval-def:TestType

Child Elements

Table 1079: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< ndd_object >

Extends: oval-def:ObjectType

Child Elements

Table 1080: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
device	oval-def:EntityObjectStringType (1..1)	The name of the device to examine.
parameter	oval-def:EntityObjectStringType (1..1)	The name of the parameter, For example, ip_forwarding.
oval-def:filter	n/a (0..unbounded)	

< ndd_state >

Extends: oval-def:StateType

Child Elements

Table 1081: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
device	oval-def:EntityStateStringType (0..1)	The name of the device to examine.
parameter	oval-def:EntityStateStringType (0..1)	The name of the parameter, For example, ip_forwarding.
value	oval-def:EntityStateAnySimpleType (0..1)	The value of the named parameter.

< patch53_test >

From /usr/sbin/swlist -l patch PHxx_yyyyy. See swlist manpage for specific fields

Extends: oval-def:TestType

Child Elements

Table 1082: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< patch53_object >

Extends: oval-def:ObjectType

Child Elements

Table 1083: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	hpux-def:Patch53Behaviors (0..1)	
swtype	oval-def:EntityObjectStringType (1..1)	HP-UX patch names begin with 'PH'
area_patched	oval-def:EntityObjectStringType (1..1)	The third and fourth characters in HP-UX patch names indicate the area of software being patched. CO - General HP-UX commands KL - Kernel patches NE - Network specific patches SS - All other subsystems (X11, starbase, etc.)
patch_base	oval-def:EntityObjectStringType (1..1)	The sixth through tenth characters in HP-UX patch names represent a unique numeric identifier for the patch
oval-def:filter	n/a (0..unbounded)	

< patch53_state >

Extends: oval-def:StateType

Child Elements

Table 1084: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
swtype	oval-def:EntityStateStringType (0..1)	HP-UX patch names begin with 'PH'
area_patched	oval-def:EntityStateStringType (0..1)	The third and fourth characters in HP-UX patch names indicate the area of software being patched. CO - General HP-UX commands KL - Kernel patches NE - Network specific patches SS - All other subsystems (X11, starbase, etc.)
patch_base	oval-def:EntityStateStringType (0..1)	The sixth through tenth characters in HP-UX patch names represent a unique numeric identifier for the patch

== Patch53Behaviors ==

The Patch53Behaviors complex type defines a number of behaviors that allow a more detailed definition of the patch53_object being specified. 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.

Attributes

Table 1085: Attributes

Attribute	Type	Desc.
supersedence	Restriction of xsd:boolean (optional *default*='false')	'supersedence' specifies that the object should also match any superseding patches to the one being specified. In other words, if set to True the resulting object set would be the original patch specified plus any superseding patches. The default value is 'false' meaning the object should only match the specified patch.

< patch_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.3
- Reason: Replaced by the patch53_test. The patch_name entity was removed from the patch_object element, and replaced with the swtype, area_patched, and patch_base entities, because the patch_name element can be constructed from the swtype, area_patched, and patch_base entities. Likewise, the patch_name entity was removed from the patch_state element for the same reason. Also, a behaviors entity was added to the patch_object to allow the object to match both the original patch and any superseding patches. A new test was created to reflect these changes. See the patch53_test.
- Comment: This test has been deprecated and will be removed in version 6.0 of the language.

From /usr/sbin/swlist -l patch PHxx_yyyyy. See swlist manpage for specific fields

Extends: oval-def:TestType

Child Elements

Table 1086: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< patch_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.3
- Reason: Replaced by the patch53_object. The patch_name entity was removed from the patch_object element, and replaced with the swtype, area_patched, and patch_base entities, because the patch_name element can be constructed from the swtype, area_patched, and patch_base entities. Also, a behaviors entity was added to the patch_object to allow the object to match both the original patch and any superseding patches. A new object was created to reflect these changes. See the patch53_object.

- Comment: This object has been deprecated and will be removed in version 6.0 of the language.

Extends: oval-def:ObjectType

Child Elements

Table 1087: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
patch_name	oval-def:EntityObjectStringType (1..1)	This is the patch name to check.

< patch_state > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.3
- Reason: Replaced by the patch53_state. The patch_name entity was removed from the patch_state element, and replaced with the swtype, area_patched, and patch_base entities, because the patch_name element can be constructed from the swtype, area_patched, and patch_base entities. A new state was created to reflect these changes. See the patch53_state.
- Comment: This state has been deprecated and will be removed in version 6.0 of the language.

Extends: oval-def:StateType

Child Elements

Table 1088: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
patch_name	oval-def:EntityStateStringType (0..1)	This is the patch name to check
swtype	oval-def:EntityStateStringType (0..1)	HP-UX patch names begin with 'PH'
area_patched	oval-def:EntityStateStringType (0..1)	The third and fourth characters in HP-UX patch names indicate the area of software being patched. CO - General HP-UX commands KL - Kernel patches NE - Network specific patches SS - All other subsystems (X11, starbase, etc.)
patch_base	oval-def:EntityStateStringType (0..1)	The sixth through tenth characters in HP-UX patch names represent a unique numeric identifier for the patch

< swlist_test >

Output of /usr/sbin/swlist command. Note: A quick way to check for the installation of a specific fileset is to use the command 'swlist -a version -l fileset filesetname'. See manpage for swlist for explanation of additional command options.

Extends: oval-def:TestType

Child Elements

Table 1089: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< swlist_object >

Extends: oval-def:ObjectType

Child Elements

Table 1090: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
swlist	oval-def:EntityObjectStringType (1..1)	This is the name of the bundle or fileset to check.
oval-def:filter	n/a (0..unbounded)	

< swlist_state >

Extends: oval-def:StateType

Child Elements

Table 1091: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
swlist	oval-def:EntityStateStringType (0..1)	This is the name of the bundle or fileset to check.
bundle	oval-def:EntityStateStringType (0..1)	
fileset	oval-def:EntityStateStringType (0..1)	
version	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	
title	oval-def:EntityStateStringType (0..1)	
vendor	oval-def:EntityStateStringType (0..1)	

< trusted_test >

This test allows for analysis of account settings in trusted HP-UX installations

Extends: oval-def:TestType

Child Elements

Table 1092: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< trusted_object >

Extends: oval-def:ObjectType

Child Elements

Table 1093: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
username	oval-def:EntityObjectStringType (1..1)	This is the name of the user being checked.
oval-def:filter	n/a (0..unbounded)	

< trusted_state >

Extends: oval-def:StateType

Child Elements

Table 1094: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
username	oval-def:EntityStateStringType (0..1)	This is the name of the user being checked
uid	oval-def:EntityStateIntType (0..1)	The user's ID
password	oval-def:EntityStateStringType (0..1)	This is the encrypted version of the user's password
account_owner	oval-def:EntityStateIntType (0..1)	The Account owner for pseudo-users
boot_auth	oval-def:EntityStateStringType (0..1)	Boot authorization
audit_id	oval-def:EntityStateStringType (0..1)	getprpwaid uses the audit ID rather than the UID
audit_flag	oval-def:EntityStateStringType (0..1)	
pw_change_min	oval-def:EntityStateStringType (0..1)	Minimum time between password changes
pw_max_size	oval-def:EntityStateIntType (0..1)	Maximum password length in characters
pw_expiration	oval-def:EntityStateIntType (0..1)	Password expiration time in seconds
pw_life	oval-def:EntityStateStringType (0..1)	Trusted lifetime, after which the account is locked
pw_change_s	oval-def:EntityStateStringType (0..1)	Time of last successful password change

Table 1094 – continued from previous page

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
pw_change_u	oval-def:EntityStateStringType (0..1)	Time of last unsuccessful password change
acct_expire	oval-def:EntityStateIntType (0..1)	Absolute account lifetime in seconds
max_llogin	oval-def:EntityStateStringType (0..1)	Maximum time allowed between logins before the account is locked
exp_warning	oval-def:EntityStateIntType (0..1)	The time in seconds before expiration when a warning will appear
usr_chg_pw	oval-def:EntityStateStringType (0..1)	Who can change this user's password
gen_pw	oval-def:EntityStateStringType (0..1)	Allows user to use system-generated passwords
pw_restrict	oval-def:EntityStateStringType (0..1)	Whether a triviality check is performed on user-generated passwords
pw_null	oval-def:EntityStateStringType (0..1)	Determines if null passwords are allowed for this account
pw_gen_char	oval-def:EntityStateStringType (0..1)	Allows password generator to use random printable ASCII characters
pw_gen_let	oval-def:EntityStateStringType (0..1)	Allows password generator to use random letters
login_time	oval-def:EntityStateStringType (0..1)	Specifies the times when the user may login to this account
pw_changer	oval-def:EntityStateIntType (0..1)	The user ID of the user who last changed the password on the user's accou
login_time_s	oval-def:EntityStateStringType (0..1)	The time of the last successful login using this account
login_time_u	oval-def:EntityStateStringType (0..1)	The time of the last unsuccessful login using this account
login_tty_s	oval-def:EntityStateStringType (0..1)	The terminal or remote host associated with the last successful login to the
login_tty_u	oval-def:EntityStateStringType (0..1)	The terminal or remote hosts associated with the last unsuccessful login to
num_u_logins	oval-def:EntityStateIntType (0..1)	The number of unsuccessful login attempts since that last successful login
max_u_logins	oval-def:EntityStateIntType (0..1)	The maximum number of unsuccessful login attempts before the account
lock_flag	oval-def:EntityStateBoolType (0..1)	Indicates whether the administrative lock on the account is set

Open Vulnerability and Assessment Language: HP-UX System Characteristics

- Schema: HP-UX System Characteristics
- Version: 5.11.1:1.1
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the HP-UX specific system characteristic items found in Open Vulnerability and Assessment Language (OVAL). Each item is an extension of the standard item element defined in the Core System Characteristic Schema. Through extension, each item inherits a set of elements and attributes that are shared amongst all OVAL Items. Each item is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core System Characteristic Schema is not outlined here.

The OVAL Schema is maintained by the 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.cisecurity.org.

Item Listing

- *< getconf_item >*
- *< ndd_item >*
- *< patch_item >*
- *< swlist_item >*
- *< trusted_item >*

< getconf_item >

These items contain getconf items.

Extends: oval-sc:ItemType

Child Elements

Table 1095: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
parameter_name	oval-sc:EntityItemStringType (0..1)	This is the parameter name to check
pathname	oval-sc:EntityItemStringType (0..1)	This is the pathname to check
output	oval-sc:EntityItemAnySimpleType (0..1)	The output produced by the getconf command.

< ndd_item >

This item represents data collected by the ndd command.

Extends: oval-sc:ItemType

Child Elements

Table 1096: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
device	oval-sc:EntityItemStringType (0..1)	The name of the device for which the parameter was collected.
parameter	oval-sc:EntityItemStringType (0..1)	The name of a parameter for example, ip_forwarding
value	oval-sc:EntityItemAnySimpleType (0..1)	The observed value of the named parameter.

< patch_item >

From /usr/sbin/swlist -l patch PHxx_yyyyy. See swlist manpage for specific fields

Extends: oval-sc:ItemType

Child Elements

Table 1097: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
patch_name	oval-sc:EntityItemStringType (0..1)	This is the patch name to check.
swtype	oval-sc:EntityItemStringType (0..1)	HP-UX patch names begin with 'PH'
area_patched	oval-sc:EntityItemStringType (0..1)	The third and fourth characters in HP-UX patch names indicate the area of software being patched. CO - General HP-UX commands KL - Kernel patches NE - Network specific patches SS - All other subsystems (X11, starbase, etc.)
patch_base	oval-sc:EntityItemStringType (0..1)	The sixth through tenth characters in HP-UX patch names represent a unique numeric identifier for the patch.

< swlist_item >

Output of /usr/sbin/swlist command. Note: A quick way to check for the installation of a specific fileset is to use the command 'swlist -a version -l fileset filesetname'. See manpage for swlist for explanation of additional command options.

Extends: oval-sc:ItemType

Child Elements

Table 1098: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
swlist	oval-sc:EntityItemStringType (0..1)	This is the name of the bundle or fileset to check.
bundle	oval-sc:EntityItemStringType (0..1)	
fileset	oval-sc:EntityItemStringType (0..1)	
version	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	
title	oval-sc:EntityItemStringType (0..1)	
vendor	oval-sc:EntityItemStringType (0..1)	

< trusted_item >

These items contain account settings for trusted HP-UX installations.

Extends: oval-sc:ItemType

Child Elements

Table 1099: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
username	oval-sc:EntityItemStringType (0..1)	This is the name of the user being checked
uid	oval-sc:EntityItemIntType (0..1)	The user’s ID
password	oval-sc:EntityItemStringType (0..1)	This is the encrypted version of the user’s password
account_owner	oval-sc:EntityItemIntType (0..1)	The Account owner for pseudo-users
boot_auth	oval-sc:EntityItemStringType (0..1)	Boot authorization
audit_id	oval-sc:EntityItemStringType (0..1)	getprpwaid uses the audit ID rather than the UID
audit_flag	oval-sc:EntityItemStringType (0..1)	
pw_change_min	oval-sc:EntityItemStringType (0..1)	Minimum time between password changes
pw_max_size	oval-sc:EntityItemIntType (0..1)	Maximum password length in characters
pw_expiration	oval-sc:EntityItemIntType (0..1)	Password expiration time in seconds
pw_life	oval-sc:EntityItemStringType (0..1)	Trusted lifetime, after which the account is locked
pw_change_s	oval-sc:EntityItemStringType (0..1)	Time of last successful password change
pw_change_u	oval-sc:EntityItemStringType (0..1)	Time of last unsuccessful password change
acct_expire	oval-sc:EntityItemIntType (0..1)	Absolute account lifetime in seconds
max_llogin	oval-sc:EntityItemStringType (0..1)	Maximum time allowed between logins before the account is locked
exp_warning	oval-sc:EntityItemIntType (0..1)	The time in seconds before expiration when a warning will appear
usr_chg_pw	oval-sc:EntityItemStringType (0..1)	Who can change this user’s password
gen_pw	oval-sc:EntityItemStringType (0..1)	Allows user to use system-generated passwords
pw_restrict	oval-sc:EntityItemStringType (0..1)	Whether a triviality check is performed on user-generated passwords
pw_null	oval-sc:EntityItemStringType (0..1)	Determines if null passwords are allowed for this account
pw_gen_char	oval-sc:EntityItemStringType (0..1)	Allows password generator to use random printable ASCII characters
pw_gen_let	oval-sc:EntityItemStringType (0..1)	Allows password generator to use random letters
login_time	oval-sc:EntityItemStringType (0..1)	Specifies the times when the user may login to this account
pw_changer	oval-sc:EntityItemIntType (0..1)	The user ID of the user who last changed the password on the user’s accour
login_time_s	oval-sc:EntityItemStringType (0..1)	The time of the last successful login using this account
login_time_u	oval-sc:EntityItemStringType (0..1)	The time of the last unsuccessful login using this account
login_tty_s	oval-sc:EntityItemStringType (0..1)	The terminal or remote host associated with the last successful login to the
login_tty_u	oval-sc:EntityItemStringType (0..1)	The terminal or remote hosts associated with the last unsuccessful login to t
num_u_logins	oval-sc:EntityItemIntType (0..1)	The number of unsuccessful login attempts since that last successful login
max_u_logins	oval-sc:EntityItemIntType (0..1)	The maximum number of unsuccessful login attempts before the account is
lock_flag	oval-sc:EntityItemBoolType (0..1)	Indicates whether the administrative lock on the account is set

Open Vulnerability and Assessment Language: AIX Definition

- Schema: AIX Definition
- Version: 5.11.1:1.1
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the AIX specific tests found in Open Vulnerability and Assessment Language (OVAL). Each test is an extension of the standard test element defined in the Core Definition Schema. Through extension, each test inherits a set of elements and attributes that are shared amongst all OVAL tests. Each test is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core Definition Schema is not outlined here.

This schema was originally developed by Yuzheng Zhou and Todd Dolinsky at Hewlett-Packard. The OVAL Schema is maintained by the 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.cisecurity.org.

Test Listing

- *< interim_fix_test >*
- *< fileset_test >*
- *< fix_test >*
- *< no_test >*
- *< oslevel_test >*

< interim_fix_test >

The interim fix test is used to check information associated with different interim or emergency fixes installed on the system. The information being tested is based off the emgr -l -u VUID command. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an interim_fix_object and the optional state element specifies the information to check.

Extends: oval-def:TestType

Child Elements

Table 1100: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< interim_fix_object >

The interim_fix_object element is used by a interim_fix_test to define the specific fix to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An interim_fix_object consists of a single vuid entity that identifies the fix to be used.

Extends: oval-def:ObjectType

Child Elements

Table 1101: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
vuid	oval-def:EntityObjectStringType (1..1)	Virtually Unique ID. A combination of time and cpuid, this ID can be used to differentiate fixes that are otherwise identical.
oval-def:filter	n/a (0..unbounded)	

< interim_fix_state >

The interim_fix_state element defines the different information associated with a specific interim fix installed on the system. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1102: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
vuid	oval-def:EntityStateStringType (0..1)	Virtually Unique ID. A combination of time and cpuid, this ID can be used to differentiate fixes that are otherwise identical.
label	oval-def:EntityStateStringType (0..1)	Each efix that is installed on a given system has a unique efix label.
abstract	oval-def:EntityStateStringType (0..1)	Describes the efix package.
state	aix-def:EntityStateInterimFixStateType (0..1)	The the emergency fix state.

< fileset_test >

The fileset_test is used to check information associated with different filesets installed on the system. The information used by this test is modeled after the /usr/bin/lslpp -l command. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an inetd_object and the optional state element specifies the information to check.

Extends: oval-def:TestType

Child Elements

Table 1103: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< fileset_object >

The fileset_object element is used by a fileset_test to define the fileset to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A fileset_object consists of a single flstinst entity that identifies the fileset to be used.

Extends: oval-def:ObjectType

Child Elements

Table 1104: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
flstinst	oval-def:EntityObjectStringType (1..1)	The flstinst entity represents the fileset name we want to check. For example, if we want to check the status of the fileset 'bos.rte', we can use fileset test and the flstinst entity will be 'bos.rte' or 'bot.*' or etc.
oval-def:filter	n/a (0..unbounded)	

< fileset_state >

The fileset_state element defines the different information associated with filesets installed on the system. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1105: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
flstinst	oval-def:EntityStateStringType (0..1)	Represents the name of a fileset.
level	oval-def:EntityStateVersionType (0..1)	Maintenance level (also known as version in Solaris or Linux) of a fileset. For example, “5.3.0.10” is the level for ‘bos.txt.tfs’ fileset in one AIX machine.
state	aix-def:EntityStateFilesetStateType (0..1)	This gives the state of a fileset. The state can be ‘APPLIED’, ‘APPLYING’,’BROKEN’, ‘COMMITTED’, ‘EFIX LOCKED’, ‘OBSOLETE’, ‘COMMITTING’,’REJECTING’. See the manpage of the ‘lslpp’ command more information.
description	oval-def:EntityStateStringType (0..1)	Short description of a fileset.

< fix_test >

The fix test is used to check information associated with different fixes installed on the system. The information being tested is based off the /usr/sbin/instfix -iavk command. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an fix_object and the optional state element specifies the information to check.

Extends: oval-def:TestType

Child Elements

Table 1106: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< fix_object >

The fix_object element is used by a fix test to define the specific fix to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A fix object consists of a single apar_number entity that identifies the fix to be used.

Extends: oval-def:ObjectType

Child Elements

Table 1107: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
apar_number	oval-def:EntityObjectStringType (1..1)	APAR is the short for 'Authorized Program Analysis Report'. APAR identifies and describes a software product defect. An APAR number can obtain a PTF (Program Temporary Fix) for the defect, if a PTF is available. An example of an apar_number is 'IY78751', it includes two alphabetic characters and a 5-digit integer.
oval-def:filter	n/a (0..unbounded)	

< fix_state >

The fix_state element defines the different information associated with a specific fix installed on the system. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1108: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
apar_number	oval-def:EntityStateStringType (0..1)	APAR is the short for 'Authorized Program Analysis Report'. APAR identifies and describes a software product defect. An APAR number can obtain a PTF (Program Temporary Fix) for the defect, if a PTF is available. An example of an apar_number is 'IY78751', it includes two alphabetic characters and a 5-digit integer.
abstract	oval-def:EntityStateStringType (0..1)	The abstract of an APAR. For instance, 'LL syas rXct are available even when not susea' is the abstract of APAR 'IY78751'.
symptom	oval-def:EntityStateStringType (0..1)	The symptom text related to an APAR. For example, the symptom text for 'IY75211' is 'Daylight savings change for year 2007 and beyond'.
installation_status	aix-def:EntityStateFixInstallationStatusType (0..1)	The installation status of files associated with the APAR. This cannot be got from the output of the instfix command directly. The last line of the output is 'All filesets for XXXXXXX were found', or 'Not all filesets for XXXXXXX were found' or 'No filesets which have fixes for XXXXXXX are currently installed.'. These can be translated to the correct value as defined by the EntityStateFixInstallationStatusType.

< no_test >

The no test is used to check information related to the /usr/sbin/no command and the parameters it manages. The no command sets or displays current or next boot values for network tuning parameters. The information being tested is based off the /usr/sbin/no -o command. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a no_object and the optional state element specifies the value to check for.

Extends: oval-def:TestType

Child Elements

Table 1109: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< no_object >

The no_object element is used by a no_test to define the specific parameter to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A no_object consists of a single tunable entity that identifies the parameter to be looked at.

Extends: oval-def:ObjectType

Child Elements

Table 1110: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
tunable	oval-def:EntityObjectStringType (1..1)	The tunable entity holds the name of the tunable parameter to be queried by the /usr/sbin/no command. Examples include ip_forwarding and tcp_keepalive_interval.
oval-def:filter	n/a (0..unbounded)	

< no_state >

The no_state element defines the different information associated with a specific call to /usr/sbin/no. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1111: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
tunable	oval-def:EntityStateStringType (0..1)	The tunable entity is used to check the name of the tunable parameter that was used by the /usr/sbin/no command. Examples include ip_forwarding and tcp_keepalive_interval.
value	oval-def:EntityStateAnySimpleType (0..1)	The value entity defines the value to check against the tunable parameter being examined.

< oslevel_test >

The oslevel test reveals information about the release and maintenance level of AIX operating system. This information can be retrieved by the /usr/bin/oslevel -r command. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an oslevel_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 1112: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< oslevel_object >

The oslevel_object element is used by an oslevel test to define those objects to be evaluated based on a specified state. There is actually only one object relating to oslevel and this is the system as a whole. Therefore, there are no child entities defined. Any OVAL Test written to check oslevel will reference the same oslevel_object which is basically an empty object element.

Extends: oval-def:ObjectType

< oslevel_state >

The oslevel_state element defines the information about maintenance level (system version). Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1113: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
maintenance_level	oval-def:EntityStateVersionType (1..1)	This is the maintenance level (system version) of current AIX operating system.

== EntityStateFilesetStateType ==

The EntityStateFilesetStateType complex type defines the different values that are valid for the state entity of a fileset state. The empty string is also allowed as a valid value to support an empty element that is found when a variable reference is used within the state entity. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 1114: Enumeration Values

Value	Description
APPLIED	The specified fileset is installed on the system. The APPLIED state means that the fileset can be rejected with the installp command and the previous level of the fileset restored. This state is only valid for Version 4 fileset updates and 3.2 migrated filesets.
APPLYING	An attempt was made to apply the specified fileset, but it did not complete successfully, and cleanup was not performed.
BROKEN	The specified fileset or fileset update is broken and should be reinstalled before being used.
COMMITTED	The specified fileset is installed on the system. The COMMITTED state means that a commitment has been made to this level of the software. A committed fileset update cannot be rejected, but a committed fileset base level and its updates (regardless of state) can be removed or deinstalled by the installp command.
COMMITTING	An attempt was made to commit the specified fileset, but it did not complete successfully, and cleanup was not performed.
EFIX LOCKED	The specified fileset was installed sucessfully and locked by the interim fix (interim fix) manager.
OBSOLETE	The specified fileset was installed with an earlier version of the operating system but has been replaced by a repackaged (renamed) newer version. Some of the files that belonged to this fileset have been replaced by versions from the repackaged fileset.
REJECTING	An attempt was made to reject the specified fileset, but it did not complete successfully, and cleanup was not performed.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateFixInstallationStatusType ==

The EntityStateFixInstallationStatusType complex type defines the different values that are valid for the installation_status entity of a fix_state state. The empty string is also allowed as a valid value to support an empty element that is found when a variable reference is used within the installation_status entity. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 1115: Enumeration Values

Value	Description
ALL_INSTALLED	All filesets for XXXXXXX were found
SOME_INSTALLED	Not all filesets for XXXXXXX were found
NONE_INSTALLED	No filesets which have fixes for XXXXXXX are currently installed.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateInterimFixStateType ==

The EntityStateInterimFixStateType complex type defines the different values that are valid for the state entity of a interim_fix_state state. Please refer to the AIX documentation of Emergency Fix States. The empty string is also allowed as a valid value to support an empty element that is found when a variable reference is used within the state entity. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 1116: Enumeration Values

Value	Description
STABLE	The efix was installed with a standard installation, and successfully completed the last installation operation.
MOUNTED	The efix was installed with a mount installation operation, and successfully completed the last installation or mount operation.
UNMOUNTED	The efix was installed with a mount installation operation and one or more efix files were unmounted in a previous emgr command operation.
BROKEN	An unrecoverable error occurred during an installation or removal operation. The status of the efix is unreliable.
INSTALLING	The efix is in the process of installing.
REBOOT_REQUIRED	The efix was installed successfully and requires a reboot to fully integrate into the target system.
REMOVING	The efix is in the process of being removed.
	The empty string value is permitted here to allow for empty elements associated with variable references.

Open Vulnerability and Assessment Language: AIX System Characteristics

- Schema: AIX System Characteristics
- Version: 5.11.1:1.1
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the AIX specific system characteristic items found in Open Vulnerability and Assessment Language (OVAL). Each item is an extension of the standard test element defined in the Core Definition Schema. Through extension, each test inherits a set of elements and attributes that are shared amongst all OVAL tests. Each test is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core Definition Schema is not outlined here.

This schema was originally developed by Yuzheng Zhou and Todd Dolinsky at Hewlett-Packard. The OVAL Schema is maintained by the 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.cisecurity.org.

Item Listing

- *< interim_fix_item >*
- *< fileset_item >*
- *< fix_item >*
- *< no_item >*
- *< oslevel_item >*

< interim_fix_item >

From emgr -l -u VUID Command. See instfix manpage for specific fields.

Extends: oval-sc:ItemType

Child Elements

Table 1117: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
vuid	oval-sc:EntityItemStringType (0..1)	Virtually Unique ID. A combination of time and cpuid, this ID can be used to differentiate fixes that are otherwise identical.
label	oval-sc:EntityItemStringType (0..1)	Each efix that is installed on a given system has a unique efix label.
abstract	oval-sc:EntityItemStringType (0..1)	Describes the efix package.
state	aix-sc:EntityItemInterimFixStateType (0..1)	The the emergency fix state.

< fileset_item >

Output of /usr/bin/lslpp -l FilesetName. See lslpp manpage for specific fields.

Extends: oval-sc:ItemType

Child Elements

Table 1118: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
flstinst	oval-sc:EntityItemStringType (0..1)	Represents the name of the fileset being checked.
level	oval-sc:EntityItemVersionType (0..1)	Maintenance level (also known as version in Solaris or Linux) of the fileset. For example, "5.3.0.10" is the level for 'bos.txt.tfs' fileset in one AIX machine.
state	aix-sc:EntityItemFilesetStateType (0..1)	This gives the state of the fileset being checked. The state can be 'APPLIED', 'APPLYING', 'BROKEN', 'COMMITTED', 'EFIX LOCKED', 'OBSOLETE', 'COMMITTING','REJECTING'. See the manpage of the 'lslpp' command more information.
description	oval-sc:EntityItemStringType (0..1)	Short description of the fileset being checked.

< fix_item >

From /usr/sbin/instfix -iavk APARNum Command. See instfix manpage for specific fields.

Extends: oval-sc:ItemType

Child Elements

Table 1119: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
apar_number	oval-sc:EntityItemStringType (0..1)	APAR is the short for 'Authorized Program Analysis Report'. APAR identifies and describes a software product defect. An APAR number can obtain a PTF (Program Temporary Fix) for the defect, if a PTF is available. An example of an apar_number is 'IY78751', it includes two alphabetic characters and a 5-digit integer.
abstract	oval-sc:EntityItemStringType (0..1)	The abstract of the APAR being checked. For instance, 'LL syas rXct are available even when not susea' is the abstract of APAR 'IY78751'.
symptom	oval-sc:EntityItemStringType (0..1)	The symptom text related to the APAR being checked. For example, the symptom text for 'IY75211' is 'Daylight savings change for year 2007 and beyond'.
installation_status	aix-sc:EntityItemFixInstallationStatusType (0..1)	The installation status of files associated with the APAR.

< no_item >

The no_item is used to hold information related to the /usr/sbin/no command and the tunable parameters it manages. Currently, /usr/sbin/no is used to configure network tuning parameters. The /usr/sbin/no command sets or displays current or next boot values for network tuning parameters. The /usr/sbin/no command queries the named parameter, retrieves the value associated with the specified parameter, and displays it.

Extends: oval-sc:ItemType

Child Elements

Table 1120: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
tunable	oval-sc:EntityItemStringType (0..1)	The name of the target parameter to be queried by the /usr/sbin/no command. Examples include ip_forwarding and tcp_keepalive_interval.
value	oval-sc:EntityItemAnySimpleType (0..1)	The value entity defines the value assigned to the tunable parameter being examined.

< oslevel_item >

Information about the release and maintenance level of AIX operating system. This information can be retrieved by the /usr/bin/oslevel -r command.

Extends: oval-sc:ItemType

Child Elements

Table 1121: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
maintenance_level	oval-sc:EntityItemVersionType (0..1)	This is the maintenance level (system version) of current AIX operating system.

== EntityItemFilesetStateType ==

The EntityStateFilesetStateType complex type defines the different values that are valid for the state entity of a fileset state. The empty string value is permitted here to allow for detailed error reporting.

Restricts: oval-sc:EntityItemStringType

Table 1122: Enumeration Values

Value	Description
APPLIED	The specified fileset is installed on the system. The APPLIED state means that the fileset can be rejected with the installp command and the previous level of the fileset restored. This state is only valid for Version 4 fileset updates and 3.2 migrated filesets.
APPLYING	An attempt was made to apply the specified fileset, but it did not complete successfully, and cleanup was not performed.
BROKEN	The specified fileset or fileset update is broken and should be reinstalled before being used.
COMMITTED	The specified fileset is installed on the system. The COMMITTED state means that a commitment has been made to this level of the software. A committed fileset update cannot be rejected, but a committed fileset base level and its updates (regardless of state) can be removed or deinstalled by the installp command.
COMMITTING	An attempt was made to commit the specified fileset, but it did not complete successfully, and cleanup was not performed.
EFIX LOCKED	The specified fileset was installed sucessfully and locked by the interim fix (interim fix) manager.
OBSOLETE	The specified fileset was installed with an earlier version of the operating system but has been replaced by a repackaged (renamed) newer version. Some of the files that belonged to this fileset have been replaced by versions from the repackaged fileset.
REJECTING	An attempt was made to reject the specified fileset, but it did not complete successfully, and cleanup was not performed.
	(No Description)

== EntityItemFixInstallationStatusType ==

The EntityStateFixInstallationStatusType defines the different values that are valid for the installation_status entity of a fix_state item. The empty string is also allowed as a valid value to support empty emlements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 1123: Enumeration Values

Value	Description
ALL_INSTALLED	All filesets for XXXXXXX were found
SOME_INSTALLED	Not all filesets for XXXXXXX were found
NONE_INSTALLED	No filesets which have fixes for XXXXXXX are currently installed.
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemInterimFixStateType ==

The EntityItemInterimFixStateType complex type defines the different values that are valid for the state entity of a interim_fix_state state. Please refer to the AIX documentation of Emergency Fix States. The empty string value is permitted here to allow for detailed error reporting.

Restricts: oval-sc:EntityItemStringType

Table 1124: Enumeration Values

Value	Description
STABLE	The efix was installed with a standard installation, and successfully completed the last installation operation.
MOUNTED	The efix was installed with a mount installation operation, and successfully completed the last installation or mount operation.
UNMOUNTED	The efix was installed with a mount installation operation and one or more efix files were unmounted in a previous emgr command operation.
BROKEN	An unrecoverable error occurred during an installation or removal operation. The status of the efix is unreliable.
INSTALLING	The efix is in the process of installing.
REBOOT_REQUIRED	The efix was installed successfully and requires a reboot to fully integrate into the target system.
REMOVING	The efix is in the process of being removed.
	The empty string value is permitted here to allow for detailed error reporting.

Open Vulnerability and Assessment Language: Linux Definition

- Schema: Linux Definition
- Version: 5.11.1:1.2
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the Linux specific tests found in Open Vulnerability and Assessment Language (OVAL). Each test is an extension of the standard test element defined in the Core Definition Schema. Through extension, each test inherits a set of elements and attributes that are shared amongst all OVAL tests. Each test is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core Definition Schema is not outlined here.

The OVAL Schema is maintained by the 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.cisecurity.org.

Test Listing

- *< apparmorstatus_test >*
- *< dpkginfo_test >*
- *< iflisteners_test >*
- *< inetlisteningservers_test > (Deprecated)*
- *< partition_test >*
- *< rpminfo_test >*
- *< rpmverify_test > (Deprecated)* (Deprecated)
- *< rpmverifyfile_test >*
- *< rpmverifypackage_test >*
- *< selinuxboolean_test >*
- *< selinuxsecuritycontext_test >*
- *< slackwarepkginfo_test >*
- *< systemdunitdependency_test >*
- *< systemdunitproperty_test >*

< apparmorstatus_test >

The AppArmor Status Test is used to check properties representing the counts of profiles and processes as per the results of the "apparmor_status" or "aa-status" command. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an apparmorstatus_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 1125: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< apparmorstatus_object >

The apparmorstatus_object element is used by an apparmorstatus test to define the different information about the current AppArmor polciy. There is actually only one object relating to AppArmor Status and this is the system as a whole. Therefore, there are no child entities defined. Any OVAL Test written to check AppArmor status will reference the same apparmorstatus_object which is basically an empty object element.

Extends: oval-def:ObjectType

< apparmorstatus_state >

The AppArmor Status Item displays various information about the current AppArmor policy. This item maps the counts of profiles and processes as per the results of the "apparmor_status" or "aa-status" command. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1126: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
loaded_profiles_count	oval-def:EntityStateIntType (0..1)	Displays the number of loaded profiles
enforce_mode_profiles_count	oval-def:EntityStateIntType (0..1)	Displays the number of profiles in enforce mode
complain_mode_profiles_count	oval-def:EntityStateIntType (0..1)	Displays the number of profiles in complain mode
processes_with_profiles_count	oval-def:EntityStateIntType (0..1)	Displays the number of processes which have profiles defined
enforce_mode_processes_count	oval-def:EntityStateIntType (0..1)	Displays the number of processes in enforce mode
complain_mode_processes_count	oval-def:EntityStateIntType (0..1)	Displays the number of processes in complain mode
unconfined_processes_with_profiles_count	oval-def:EntityStateIntType (0..1)	Displays the number of processes which are unconfined but have a profile defined

< dpkginfo_test >

The dpkginfo test is used to check information for a given DPKG package. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a dpkginfo_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 1127: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< dpkginfo_object >

The dpkginfo_object element is used by a dpkginfo test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A dpkginfo object consists of a single name entity that identifies the package being checked.

Extends: oval-def:ObjectType

Child Elements

Table 1128: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityObjectStringType (1..1)	This is the package name to check.
oval-def:filter	n/a (0..unbounded)	

< dpkginfo_state >

The dpkginfo_state element defines the different information that can be used to evaluate the specified DPKG package. This includes the architecture, epoch number, release, and version numbers. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1129: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	This is the DPKG package name to check.
arch	oval-def:EntityStateStringType (0..1)	This is the architecture for which the package was built, like : i386, ppc, sparc, noarch.
epoch	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	This is the epoch number of the DPKG. For a null epoch (or '(none)' as returned by dpkg) the string '(none)' should be used.
release	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	This is the release number of the build, changed by the vendor/builder.
version	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	This is the version number of the build.
evr	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	This represents the epoch, upstream_version, and debian_revision fields, for a Debian package, as a single version string. It has the form "EPOCH:UPSTREAM_VERSION-DEBIAN_REVISION". Note that a null epoch (or '(none)' as returned by dpkg) is equivalent to '0' and would hence have the form 0:UPSTREAM_VERSION-DEBIAN_REVISION.

< iflisteners_test >

The iflisteners_test is used to check what applications such as packet sniffers that are bound to an interface on the system. This is limited to applications that are listening on AF_PACKET sockets. Furthermore, only applications bound to an ethernet interface should be collected. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an iflisteners_object and the optional iflisteners_state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 1130: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< iflisteners_object >

The iflisteners_object element is used by an iflisteners_test to define the specific interface to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 1131: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
interface_name	oval-def:EntityObjectStringType (1..1)	The interface_name entity specifies the name of the interface (eth0, eth1, fw0, etc.) to check.
oval-def:filter	n/a (0..unbounded)	

< iflisteners_state >

The iflisteners_state element defines the different information that can be used to evaluate the specified applications that are listening on interfaces on the system. This includes the interface name, protocol, hardware address, program name, pid, and user id. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1132: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
interface_name	oval-def:EntityStateStringType (0..1)	This is the name of the interface (eth0, eth1, fw0, etc.).
protocol	linux-def:EntityStateProtocolType (0..1)	This is the physical layer protocol used by the AF_PACKET socket.
hw_address	oval-def:EntityStateStringType (0..1)	This is the hardware address associated with the interface.
program_name	oval-def:EntityStateStringType (0..1)	This is the name of the communicating program.
pid	oval-def:EntityStateIntType (0..1)	The pid is the process ID of a specific process.
user_id	oval-def:EntityStateIntType (0..1)	The numeric user id, or uid, is the third column of each user's entry in /etc/passwd. It represents the owner, and thus privilege level, of the specified program.

< inetlisteningservers_test >

The inet listening servers test is used to check what applications are listening on the network. This is limited to applications that are listening for connections that use the TCP or UDP protocols and have addresses represented as IPv4 or IPv6 addresses (AF_INET or AF_INET6). It is generally using the parsed output of running the command netstat -tuwlnpe with root privilege. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an inetlisteningservers_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 1133: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< inetlisteningservers_object >

The inetlisteningservers_object element is used by an inet listening servers test to define the specific protocol-address-port to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An inet listening servers object consists of three entities. The first identifies a specific IP address. The second entity represents a certain port number. While the third identifies the protocol.

Extends: oval-def:ObjectType

Child Elements

Table 1134: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
protocol	oval-def:EntityObjectStringType (1..1)	The protocol entity defines a certain transport-layer protocol, in lowercase: tcp or udp.
local_address	oval-def:EntityObjectIPAddressStringType (1..1)	This is the IP address of the network interface on which an application listens. Note that the IP address can be IPv4 or IPv6.
local_port	oval-def:EntityObjectIntType (1..1)	This is the TCP or UDP port on which an application would listen. Note that this is not a list – if a program listens on multiple ports, or on a combination of TCP and UDP, each will be represented by its own object.
oval-def:filter	n/a (0..unbounded)	

< inetlisteningservers_state >

The inetlisteningservers_state element defines the different information that can be used to evaluate the specified inet listening server. This includes the local address, foreign address, port information, and process id. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1135: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
protocol	oval-def:EntityStateStringType (0..1)	The protocol entity defines the specific transport-layer protocol, in lowercase: tcp or udp, associated with the inet listening server.
local_address	oval-def:EntityStateIPAddressStringType (0..1)	This is the IP address of the network interface on which the program listens. Note that the IP address can be IPv4 or IPv6.
local_port	oval-def:EntityStateIntType (0..1)	This is the TCP or UDP port number associated with the inet listening server.
local_full_address	oval-def:EntityStateStringType (0..1)	This is the IP address and network port number associated with the inet listening server, equivalent to local_address:local_port. Note that the IP address can be IPv4 or IPv6.
program_name	oval-def:EntityStateStringType (0..1)	This is the name of the communicating program.
foreign_address	oval-def:EntityStateIPAddressStringType (0..1)	This is the IP address with which the program is communicating, or with which it will communicate, in the case of a listening server. Note that the IP address can be IPv4 or IPv6.
foreign_port	oval-def:EntityStateIntType (0..1)	This is the TCP or UDP port to which the program communicates. In the case of a listening program accepting new connections, the value will be 0.
foreign_full_address	oval-def:EntityStateStringType (0..1)	This is the IP address and network port to which the program is communicating or will accept communications from, equivalent to foreign_address:foreign_port. Note that the IP address can be IPv4 or IPv6.
pid	oval-def:EntityStateIntType (0..1)	The pid is the process ID of a specific process.
user_id	oval-def:EntityStateIntType (0..1)	The numeric user id, or uid, is the third column of each user's entry in /etc/passwd. It represents the owner, and thus privilege level, of the specified program.

< partition_test >

The partition_test is used to check the information associated with partitions on the local system. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a partition_object and the optional state element references a partition_state that specifies the information to check.

Extends: oval-def:TestType

Child Elements

Table 1136: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< partition_object >

The partition_object is used by a partition_test to define which partitions on the local system should be collected. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 1137: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
mount_point	oval-def:EntityObjectStringType (1..1)	The mount_point element specifies the mount points of the partitions that should be collected from the local system.
oval-def:filter	n/a (0..unbounded)	

< partition_state >

The partition_state element defines the different information associated with a partition. This includes the name, filesystem type, mount options, total space, space used, and space left. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1138: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
mount_point	oval-def:EntityStateStringType (0..1)	The mount_point element contains a string that represents the mount point of a partition on the local system.
device	oval-def:EntityStateStringType (0..1)	The device element contains a string that represents the name of the device.
uuid	oval-def:EntityStateStringType (0..1)	The uuid element contains a string that represents the universally unique identifier associated with a partition.
fs_type	oval-def:EntityStateStringType (0..1)	The fs_type element contains a string that represents the type of filesystem on a partition.
mount_options	oval-def:EntityStateStringType (0..1)	The mount_options element contains a string that represents the mount options associated with a partition. Implementation note: not all mount options are visible in /etc/mtab or /proc/mounts. A complete source of additional mount options is the f_flag field of 'struct statvfs'. See statvfs(2). /etc/fstab may have additional mount options, but it need not contain all mounted filesystems, so it MUST NOT be relied upon. Implementers MUST be sure to get all mount options in some way.
total_space	oval-def:EntityStateIntType (0..1)	The total_space element contains an integer that represents the total number of physical blocks on a partition.
space_used	oval-def:EntityStateIntType (0..1)	The space_used element contains an integer that represents the number of physical blocks used on a partition.
space_left	oval-def:EntityStateIntType (0..1)	The space_left element contains an integer that represents the number of physical blocks left on a partition available to be used by privileged users.
space_left_for_unprivileged_users	oval-def:EntityStateIntType (0..1)	The space_left_for_unprivileged_users element contains an integer that represents the number of physical blocks remaining on a partition that are available to be used by unprivileged users.
block_size	oval-def:EntityStateIntType (0..1)	The block_size element contains an integer that represents the actual byte size of each physical block on the partition's block device. This is the same block size used to compute the total_space, space_used, and space_left.

< rpminfo_test >

The rpminfo_test is used to check the RPM header information for a given RPM package. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a rpminfo_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 1139: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< rpminfo_object >

The rpminfo_object element is used by a rpm info test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A rpm info object consists of a single name entity that identifies the package being checked.

Extends: oval-def:ObjectType

Child Elements

Table 1140: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	linux-def:RpmInfoBehaviors (0..1)	
name	oval-def:EntityObjectStringType (1..1)	This is the package name to check.
oval-def:filter	n/a (0..unbounded)	

< rpminfo_state >

The rpminfo_state element defines the different information that can be used to evaluate the specified rpm. This includes the architecture, epoch number, and version numbers. Most of this information can be obtained through the rpm function. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1141: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	This is the package name to check.
arch	oval-def:EntityStateStringType (0..1)	This is the architecture for which the RPM was built, like : i386, ppc, sparc, noarch. In the case of an apache rpm named httpd-2.0.40-21.11.4.i686.rpm, this value would be i686.
epoch	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	This is the epoch number of the RPM, this is used as a kludge for version-release comparisons where the vendor has done some kind of re-numbering or version forking. For a null epoch (or '(none)' as returned by rpm) the string '(none)' should be used.. This number is not revealed by a normal query of the RPM's information – you must use a formatted rpm query command to gather this data from the command line, like so. For an already-installed RPM: rpm -q –qf '%{EPOCH}n' installed_rpm For an RPM file that has not been installed: rpm -qp –qf '%{EPOCH}n' rpm_file
release	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	This is the release number of the build, changed by the vendor/builder.
version	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	This is the version number of the build. In the case of an apache rpm named httpd-2.0.40-21.11.4.i686.rpm, this value would be 2.0.40.
evr	oval-def:EntityStateEVRStringType (0..1)	This represents the epoch, version, and release fields as a single version string. It has the form "EPOCH:VERSION-RELEASE". Note that a null epoch (or '(none)' as returned by rpm) is equivalent to '0' and would hence have the form 0:VERSION-RELEASE. Comparisons involving this datatype should follow the algorithm of librpm's rpmvercmp() function.
signature_keyid	oval-def:EntityStateStringType (0..1)	This field contains the 64-bit PGP key ID that the RPM issuer (generally the original operating system vendor) uses to sign the key. Note that the value should NOT contain a hyphen to separate the higher 32-bits from the lower 32-bits. It should simply be a 16 character hex string. PGP is used to verify the authenticity and integrity of the RPM being considered. Software packages and patches are signed cryptographically to allow administrators to allay concerns that the distribution mechanism has been compromised, whether that mechanism is web site, FTP server, or even a mirror controlled by a hostile party. OVAL uses this field most of all to confirm that the package installed on the system is that shipped by the vendor, since comparing package version numbers against patch announcements is only programmatically valid if the installed package is known to contain the patched code.
extended_name	oval-def:EntityStateStringType (0..1)	This represents the name, epoch, version, release, and architecture fields as a single version string. It has the form "NAME-EPOCH:VERSION-RELEASE.ARCHITECTURE". Note that a null epoch (or '(none)' as returned by rpm) is equivalent to '0' and would hence have the form NAME-0:VERSION-RELEASE.ARCHITECTURE. The 'gpg-pubkey' virtual package on RedHat and CentOS should use the string '(none)' for the architecture to construct the ex-

== RpmInfoBehaviors ==

The RpmInfoBehaviors complex type defines a set of behaviors for controlling what data, for installed rpms, is collected. This behavior aligns with the rpm command.

Attributes

Table 1142: Attributes

Attribute	Type	Desc.
filepaths	xsd:boolean (optional *default*='false')	'filepaths', when true, this behavior means collect all filepaths (directory and file information) from the rpm database for the package.

< rpmverify_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.10
- Reason: Replaced by the rpmverifyfile_test and the rpmverifypackage_test. The rpmverify_test was split into two tests to distinguish between the verification of the files in an rpm and the verification of an rpm as a whole. By making this distinction, content authoring is simplified and information is no longer duplicated across items. See the rpmverifyfile_test and rpmverifypackage_test.
- Comment: This test has been deprecated and will be removed in version 6.0 of the language.

The rpmverify_test is used to verify the integrity of installed RPMs. This test aligns with the rpm -V command for verifying RPMs. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a rpmverify_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 1143: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< rpmverify_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.10

- Reason: Replaced by the rpmverifyfile_object and rpmverifypackage_object. The rpmverify_test was split into two tests to distinguish between the verification of the files in an rpm and the verification of an rpm as a whole. By making this distinction, content authoring is simplified and information is no longer duplicated across items. See the rpmverifyfile_object and rpmverifypackage_object.
- Comment: This object has been deprecated and will be removed in version 6.0 of the language.

The rpmverify_object element is used by a rpmverify_test to define a set of files within a set of RPMs to verify. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 1144: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	linux-def:RpmVerifyBehaviors (0..1)	
name	oval-def:EntityObjectStringType (1..1)	This is the package name to check.
filepath	oval-def:EntityObjectStringType (1..1)	The filepath element specifies the absolute path for a file or directory in the specified package.
oval-def:filter	n/a (0..unbounded)	

< rpmverify_state > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.10
- Reason: Replaced by the rpmverifyfile_state and rpmverifypackage_state. The rpmverify_test was split into two tests to distinguish between the verification of the files in an rpm and the verification of an rpm as a whole. By making this distinction, content authoring is simplified and information is no longer duplicated across items. See the rpmverifyfile_state and rpmverifypackage_state.
- Comment: This state has been deprecated and will be removed in version 6.0 of the language.

The rpmverify_state element defines the different information that can be used to evaluate the specified rpm. This includes the architecture, epoch number, and version numbers. Most of this information can be obtained through the rpm function. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1145: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	This is the package name to check.
filepath	oval-def:EntityStateStringType (0..1)	The filepath element specifies the absolute path for a file or directory in the specified package.
size_differs	linux-def:EntityStateRpmVerifyResultType (0..1)	The size_differs entity aligns with the first character ('S' flag) in the character string in the output generated by running rpm –V on a specific file.
mode_differs	linux-def:EntityStateRpmVerifyResultType (0..1)	The mode_differs entity aligns with the second character ('M' flag) in the character string in the output generated by running rpm –V on a specific file.
md5_differs	linux-def:EntityStateRpmVerifyResultType (0..1)	The md5_differs entity aligns with the third character ('5' flag) in the character string in the output generated by running rpm –V on a specific file.
device_differs	linux-def:EntityStateRpmVerifyResultType (0..1)	The device_differs entity aligns with the fourth character ('D' flag) in the character string in the output generated by running rpm –V on a specific file.
link_mismatch	linux-def:EntityStateRpmVerifyResultType (0..1)	The link_mismatch entity aligns with the fifth character ('L' flag) in the character string in the output generated by running rpm –V on a specific file.
ownership_differs	linux-def:EntityStateRpmVerifyResultType (0..1)	The ownership_differs entity aligns with the sixth character ('U' flag) in the character string in the output generated by running rpm –V on a specific file.
group_differs	linux-def:EntityStateRpmVerifyResultType (0..1)	The group_differs entity aligns with the seventh character ('U' flag) in the character string in the output generated by running rpm –V on a specific file.
mtime_differs	linux-def:EntityStateRpmVerifyResultType (0..1)	The mtime_differs entity aligns with the eighth character ('T' flag) in the character string in the output generated by running rpm –V on a specific file.
capabilities_differ	linux-def:EntityStateRpmVerifyResultType (0..1)	The size_differs entity aligns with the ninth character ('P' flag) in the character string in the output generated by running rpm –V on a specific file.
configuration_file	oval-def:EntityStateBoolType (0..1)	The configuration_file entity represents the configuration file attribute marker that may be present on a file.
documentation_file	oval-def:EntityStateBoolType (0..1)	The documentation_file entity represents the documenation file attribute marker that may be present on a file.
ghost_file	oval-def:EntityStateBoolType (0..1)	The ghost_file entity represents the ghost file attribute marker that may be present on a file.
license_file	oval-def:EntityStateBoolType (0..1)	The license_file entity represents the license file attribute marker that may be present on a file.
readme_file	oval-def:EntityStateBoolType (0..1)	The readme_file entity represents the readme file attribute marker that may be present on a file.

== RpmVerifyBehaviors == (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.10
- Reason: Replaced by the RpmVerifyFileBehaviors and the RpmVerifyPackageBehaviors. The RpmVerifyBehaviors complex type is used by the rpmverify_test which was split into two tests to distinguish between the verification of the files in an rpm and the verification of an rpm as a whole. By making this distinction, content authoring is simplified and information is no longer duplicated across items. The new tests utilize the RpmVerifyFileBehaviors and RpmVerifyPackageBehaviors complex types, and as a result, the RpmVerifyBehaviors complex type is no longer needed.
- Comment: This complex type has been deprecated and will be removed in version 6.0 of the language.

The RpmVerifyBehaviors complex type defines a set of behaviors that for controlling how installed rpms are verified. These behaviors align with the verify-options of the rpm command with the addition of two behaviors that will indicate that a file with a given attribute marker should not be collected.

Attributes

Table 1146: Attributes

Attribute	Type	Desc.
nodeps	xsd:boolean (optional *default*='false')	'nodeps' when true this behavior means, don't verify dependencies of packages.
nodigest	xsd:boolean (optional *default*='false')	'nodigest' when true this behavior means, don't verify package or header digests when reading.
nofiles	xsd:boolean (optional *default*='false')	'nofiles' when true this behavior means, don't verify any attributes of package files.
noscripts	xsd:boolean (optional *default*='false')	'noscripts' when true this behavior means, don't execute the %verifyscript scriptlet (if any).
nosignature	xsd:boolean (optional *default*='false')	'nosignature' when true this behavior means, don't verify package or header signatures when reading.
nolinkto	xsd:boolean (optional *default*='false')	'nolinkto' when true this behavior means, don't verify symbolic links attribute.
nomd5	xsd:boolean (optional *default*='false')	'nomd5' when true this behavior means, don't verify the file md5 attribute.
nosize	xsd:boolean (optional *default*='false')	'nosize' when true this behavior means, don't verify the file size attribute.
nouser	xsd:boolean (optional *default*='false')	'nouser' when true this behavior means, don't verify the file owner attribute.
nogroup	xsd:boolean (optional *default*='false')	'nogroup' when true this behavior means, don't verify the file group owner attribute.
nomtime	xsd:boolean (optional *default*='false')	'nomtime' when true this behavior means, don't verify the file mtime attribute.
nomode	xsd:boolean (optional *default*='false')	'nomode' when true this behavior means, don't verify the file mode attribute.
nordev	xsd:boolean (optional *default*='false')	'nordev' when true this behavior means, don't verify the file rdev attribute.
noconfigfiles	xsd:boolean (optional *default*='false')	'noconfigfiles' when true this behavior means, skip files that are marked with the %config attribute marker.
noghostfiles	xsd:boolean (optional *default*='false')	'noghostfiles' when true this behavior means, skip files that are maked with %ghost attribute marker.

< rpmverifyfile_test >

The rpmverifyfile_test is used to verify the integrity of the individual files in installed RPMs. This test aligns with the rpm -V command for verifying RPMs. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a rpmverifyfile_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 1147: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< rpmverifyfile_object >

The rpmverifyfile_object element is used by a rpmverifyfile_test to define a set of files within a set of RPMs to verify. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 1148: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	linux-def:RpmVerifyFileBehaviors (0..1)	
name	oval-def:EntityObjectStringType (1..1)	This is the package name to check.
epoch	Restriction of oval-def:EntityObjectAnySimpleType. See schema for details. (1..1)	This is the epoch number of the RPM, this is used as a kludge for version-release comparisons where the vendor has done some kind of re-numbering or version forking. For a null epoch (or '(none)' as returned by rpm) the string '(none)' should be used.. This number is not revealed by a normal query of the RPM's information – you must use a formatted rpm query command to gather this data from the command line, like so. For an already-installed RPM: rpm -q –qf '%{EPOCH}n' installed_rpm For an RPM file that has not been installed: rpm -qp –qf '%{EPOCH}n' rpm_file
version	Restriction of oval-def:EntityObjectAnySimpleType. See schema for details. (1..1)	This is the version number of the build. In the case of an apache rpm named httpd-2.0.40-21.11.4.i686.rpm, this value would be 2.0.40.
release	Restriction of oval-def:EntityObjectAnySimpleType. See schema for details. (1..1)	This is the release number of the build, changed by the vendor/builder.
arch	oval-def:EntityObjectStringType (1..1)	This is the architecture for which the RPM was built, like : i386, ppc, sparc, noarch. In the case of an apache rpm named httpd-2.0.40-21.11.4.i686.rpm, this value would be i686.
filepath	oval-def:EntityObjectStringType (1..1)	The filepath element specifies the absolute path for a file or directory in the specified package.
oval-def:filter	n/a (0..unbounded)	

< rpmverifyfile_state >

The rpmverifyfile_state element defines the different information that can be used to determine if a set of files within a set of RPMs passed verification. This includes the architecture, epoch number, version numbers, and the verification of various file attributes. Most of this information can be obtained through the rpm function. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1149: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	This is the package name to check.
epoch	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	This is the epoch number of the RPM, this is used as a kludge for version-release comparisons where the vendor has done some kind of re-numbering or version forking. For a null epoch (or '(none)' as returned by rpm) the string '(none)' should be used.. This number is not revealed by a normal query of the RPM's information – you must use a formatted rpm query command to gather this data from the command line, like so. For an already-installed RPM: rpm -q –qf '%{EPOCH}n' installed_rpm For an RPM file that has not been installed: rpm -qp –qf '%{EPOCH}n' rpm_file
version	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	This is the version number of the build. In the case of an apache rpm named httpd-2.0.40-21.11.4.i686.rpm, this value would be 2.0.40.
release	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	This is the release number of the build, changed by the vendor/builder.
arch	oval-def:EntityStateStringType (0..1)	This is the architecture for which the RPM was built, like : i386, ppc, sparc, noarch. In the case of an apache rpm named httpd-2.0.40-21.11.4.i686.rpm, this value would be i686.
filepath	oval-def:EntityStateStringType (0..1)	The filepath element specifies the absolute path for a file or directory in the specified package.
extended_name	oval-def:EntityStateStringType (0..1)	This represents the name, epoch, version, release, and architecture fields as a single version string. It has the form "NAME-EPOCH:VERSION-RELEASE.ARCHITECTURE". Note that a null epoch (or '(none)' as returned by rpm) is equivalent to '0' and would hence have the form NAME-0:VERSION-RELEASE.ARCHITECTURE.
size_differs	linux-def:EntityStateRpmVerifyResultType (0..1)	The size_differs entity aligns with the first character ('S' flag) in the character string in the output generated by running rpm –V on a specific file.
mode_differs	linux-def:EntityStateRpmVerifyResultType (0..1)	The mode_differs entity aligns with the second character ('M' flag) in the character string in the output generated by running rpm –V on a specific file.
md5_differs (Deprecated)	linux-def:EntityStateRpmVerifyResultType (0..1)	The md5_differs entity aligns with the third character ('5' flag) in the character string in the output generated by running rpm –V on a specific file.
filedigest_differs	linux-def:EntityStateRpmVerifyResultType (0..1)	The filedigest_differs entity aligns with the third character ('5' flag) in the character string in the output generated by running rpm –V on a specific file. This replaces the md5_differs entity due to naming changes for verification and reporting options.
device_differs	linux-def:EntityStateRpmVerifyResultType (0..1)	The device_differs entity aligns with the fourth character ('D' flag) in the character string in the output generated by running rpm –V on a specific file.
link_mismatch	linux-def:EntityStateRpmVerifyResultType (0..1)	The link_mismatch entity aligns with the fifth character ('L' flag) in the character string in the output generated by running rpm –V on a specific file.
own	linux-	The ownership_differs entity aligns with the sixth character ('U' flag) in the character string

== RpmVerifyFileBehaviors ==

The RpmVerifyFileBehaviors complex type defines a set of behaviors that for controlling how the individual files in installed rpms are verified. These behaviors align with the verify-options of the rpm command with the addition of two behaviors that will indicate that a file with a given attribute marker should not be collected.

Attributes

Table 1150: Attributes

Attribute	Type	Desc.
nolinkto	xsd:boolean (optional *default*='false')	'nolinkto' when true this behavior means, don't verify symbolic links attribute.
nomd5 (Deprecated)	xsd:boolean (optional *default*='false')	'nomd5' when true this behavior means, don't verify the file md5 attribute.
nosize	xsd:boolean (optional *default*='false')	'nosize' when true this behavior means, don't verify the file size attribute.
nouser	xsd:boolean (optional *default*='false')	'nouser' when true this behavior means, don't verify the file owner attribute.
nogroup	xsd:boolean (optional *default*='false')	'nogroup' when true this behavior means, don't verify the file group owner attribute.
nomtime	xsd:boolean (optional *default*='false')	'nomtime' when true this behavior means, don't verify the file mtime attribute.
nomode	xsd:boolean (optional *default*='false')	'nomode' when true this behavior means, don't verify the file mode attribute.
nordev	xsd:boolean (optional *default*='false')	'nordev' when true this behavior means, don't verify the file rdev attribute.
noconfigfiles	xsd:boolean (optional *default*='false')	'noconfigfiles' when true this behavior means, skip files that are marked with the %config attribute marker.
noghostfiles	xsd:boolean (optional *default*='false')	'noghostfiles' when true this behavior means, skip files that are maked with %ghost attribute marker.
nofiledigest	xsd:boolean (optional *default*='false')	'nofiledigest' when true this behavior means, don't verify the file digest attribute.
nocaps	xsd:boolean (optional *default*='false')	'nocaps' when true this behavior means, don't verify the presence of file capabilities.

< rpmverifypackage_test >

The rpmverifypackage_test is used to verify the integrity of installed RPMs. This test aligns with the rpm -V command for verifying RPMs. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a rpmverifypackage_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 1151: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< rpmverifypackage_object >

The rpmverifypackage_object element is used by a rpmverify_test to define a set of RPMs to verify. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 1152: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	linux-def:RpmVerifyPackageBehaviors (0..1)	
name	oval-def:EntityObjectStringType (1..1)	This is the package name to check.
epoch	Restriction of oval-def:EntityObjectAnySimpleType. See schema for details. (1..1)	This is the epoch number of the RPM, this is used as a kludge for version-release comparisons where the vendor has done some kind of re-numbering or version forking. For a null epoch (or '(none)' as returned by rpm) the string '(none)' should be used.. This number is not revealed by a normal query of the RPM's information – you must use a formatted rpm query command to gather this data from the command line, like so. For an already-installed RPM: rpm -q –qf '%{EPOCH}n' installed_rpm For an RPM file that has not been installed: rpm -qp –qf '%{EPOCH}n' rpm_file
version	Restriction of oval-def:EntityObjectAnySimpleType. See schema for details. (1..1)	This is the version number of the build. In the case of an apache rpm named httpd-2.0.40-21.11.4.i686.rpm, this value would be 2.0.40.
release	Restriction of oval-def:EntityObjectAnySimpleType. See schema for details. (1..1)	This is the release number of the build, changed by the vendor/builder.
arch	oval-def:EntityObjectStringType (1..1)	This is the architecture for which the RPM was built, like : i386, ppc, sparc, noarch. In the case of an apache rpm named httpd-2.0.40-21.11.4.i686.rpm, this value would be i686.
oval-def:filter	n/a (0..unbounded)	

< rpmverifypackage_state >

The rpmverifypackage_state element defines the different information that can be used to verify the integrity of installed rpms. This includes the architecture, epoch number, version numbers, verification of variuos attributes of an rpm. Most of this information can be obtained through the rpm function. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1153: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	This is the package name to check.
epoch	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	This is the epoch number of the RPM, this is used as a kludge for version-release comparisons where the vendor has done some kind of re-numbering or version forking. For a null epoch (or '(none)' as returned by rpm) the string '(none)' should be used.. This number is not revealed by a normal query of the RPM's information – you must use a formatted rpm query command to gather this data from the command line, like so. For an already-installed RPM: rpm -q –qf '%{EPOCH}n' installed_rpm For an RPM file that has not been installed: rpm -qp –qf '%{EPOCH}n' rpm_file
version	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	This is the version number of the build. In the case of an apache rpm named httpd-2.0.40-21.11.4.i686.rpm, this value would be 2.0.40.
release	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	This is the release number of the build, changed by the vendor/builder.
arch	oval-def:EntityStateStringType (0..1)	This is the architecture for which the RPM was built, like : i386, ppc, sparc, noarch. In the case of an apache rpm named httpd-2.0.40-21.11.4.i686.rpm, this value would be i686.
extended_name	oval-def:EntityStateStringType (0..1)	This represents the name, epoch, version, release, and architecture fields as a single version string. It has the form "NAME-EPOCH:VERSION-RELEASE.ARCHITECTURE". Note that a null epoch (or '(none)' as returned by rpm) is equivalent to '0' and would hence have the form NAME-0:VERSION-RELEASE.ARCHITECTURE.
dependency_check_passed	oval-def:EntityStateBoolType (0..1)	The dependency_check_passed entity indicates whether or not the dependency check passed. If the dependency check is not performed, due to the 'nodeps' behavior, this entity must not be collected.
digest_check_passed (Deprecated)	oval-def:EntityStateBoolType (0..1)	The digest_check_passed entity indicates whether or not the verification of the package or header digests passed. If the digest check is not performed, due to the 'nodigest' behavior, this entity must not be collected.
verification_script_successful	oval-def:EntityStateBoolType (0..1)	The verification_script_successful entity indicates whether or not the verification script executed successfully. If the verification script is not executed, due to the 'noscripts' behavior, this entity must not be collected.
signature_check_passed (Deprecated)	oval-def:EntityStateBoolType (0..1)	The signature_check_passed entity indicates whether or not the verification of the package or header signatures passed. If the signature check is not performed, due to the 'nosignature' behavior, this entity must not be collected.

== RpmVerifyPackageBehaviors ==

The RpmVerifyPackageBehaviors complex type defines a set of behaviors that for controlling how installed rpms are verified. These behaviors align with the verify-options of the rpm command.

Attributes

Table 1154: Attributes

Attribute	Type	Desc.
nodeps	xsd:boolean (optional *default*='false')	'nodeps' when true this behavior means, don't verify dependencies of packages.
nodigest (Deprecated)	xsd:boolean (optional *default*='false')	'nodigest' when true this behavior means, don't verify package or header digests when reading.
noscripts	xsd:boolean (optional *default*='false')	'noscripts' when true this behavior means, don't execute the %verifyscript scriptlet (if any).
nosignature (Deprecated)	xsd:boolean (optional *default*='false')	'nosignature' when true this behavior means, don't verify package or header signatures when reading.

< selinuxboolean_test >

The selinuxboolean_test is used to check the current and pending status of a SELinux boolean. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a selinuxboolean_object and the optional state element references a selinuxboolean_state that specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 1155: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< selinuxboolean_object >

The selinuxboolean_object element is used by an selinuxboolean_test to define the items to evaluate based on a specified state.

Extends: oval-def:ObjectType

Child Elements

Table 1156: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityObjectStringType (1..1)	The name of the SELinux boolean.
oval-def:filter	n/a (0..unbounded)	

< selinuxboolean_state >

The selinuxboolean_state element defines the different information that can be used to evaluate the specified SELinux boolean. This includes SELinux boolean's current and pending status. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1157: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	The name of the SELinux boolean.
current_status	oval-def:EntityStateBoolType (0..1)	The current_status entity represents the current state of the specified SELinux boolean.
pending_status	oval-def:EntityStateBoolType (0..1)	The pending_status entity represents the pending state of the specified SELinux boolean.

< selinuxsecuritycontext_test >

The selinuxsecuritycontext_test is used to check the security context of a file or process on the local system. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a selinuxsecuritycontext_object and the optional state element references a selinuxsecuritycontext_state that specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 1158: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< selinuxsecuritycontext_object >

The selinuxsecuritycontext_object element is used by an selinuxsecuritycontext_test to define the security contexts of files and processes to collect from the local system. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 1159: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	linux-def:FileBehaviors (0..1)	
filepath	oval-def:EntityObjectStringType (1..1)	The filepath element specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath.
path	oval-def:EntityObjectStringType (1..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-def:EntityObjectStringType (1..1)	The filename element specifies the name of a file to evaluate. If the xsi:nil attribute is set to true, then the object being specified is the higher level directory object (not all the files in the directory). In this case, the filename element should not be used during collection and would result in the unique set of items being the directories themselves. For example, one would set xsi:nil to true if the desire was to test the attributes or permissions associated with a directory. Setting xsi:nil equal to true is different than using a .* pattern match, which says to collect every file under a given path.
pid	oval-def:EntityObjectIntType (1..1)	The pid entity is the process ID of the process. If the xsi:nil attribute is set to true, the process ID shall be the tool's running process.
oval-def:filter	n/a (0..unbounded)	

< selinuxsecuritycontext_state >

The selinuxsecuritycontext_state element defines the different information that can be used to evaluate the specified SELinux security context. This includes SELinux security context's user, type role, low sensitivity, low category, high sensitivity, high category, raw low sensitivity, raw low category, raw high sensitivity, and raw high category. This state follows the SELinux security context structure: user:role:type:low_sensitivity[:low_category]- high_sensitivity [:high_category]. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1160: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-def:EntityStateStringType (0..1)	The filepath element specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath.
path	oval-def:EntityStateStringType (0..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-def:EntityStateStringType (0..1)	The name of the file. If the xsi:nil attribute is set to true, then the item being represented is the higher directory represented by the path entity.
pid	oval-def:EntityStateIntType (0..1)	This is the process ID of the process.
user	oval-def:EntityStateStringType (0..1)	The user element specifies the SELinux user that either created the file or started the process.
role	oval-def:EntityStateStringType (0..1)	The role element specifies the types that a process may transition to (domain transitions). Note that this entity is not relevant for files and will always have a value of object_r.
type	oval-def:EntityStateStringType (0..1)	The type element specifies the domain in which the file is accessible or the domain in which a process executes.
low_sensitivity	oval-def:EntityStateStringType (0..1)	The low_sensitivity element specifies the current sensitivity of a file or process.
low_category	oval-def:EntityStateStringType (0..1)	The low_category element specifies the set of categories associated with the low sensitivity.
high_sensitivity	oval-def:EntityStateStringType (0..1)	The high_sensitivity element specifies the maximum range for a file or the clearance for a process.
high_category	oval-def:EntityStateStringType (0..1)	The high_category element specifies the set of categories associated with the high sensitivity.
rawlow_sensitivity	oval-def:EntityStateStringType (0..1)	The rawlow_sensitivity element specifies the current sensitivity of a file or process but in its raw context.
rawlow_category	oval-def:EntityStateStringType (0..1)	The rawlow_category element specifies the set of categories associated with the low sensitivity but in its raw context.
rawhigh_sensitivity	oval-def:EntityStateStringType (0..1)	The rawhigh_sensitivity element specifies the maximum range for a file or the clearance for a process but in its raw context.
rawhigh_category	oval-def:EntityStateStringType (0..1)	The rawhigh_category element specifies the set of categories associated with the high sensitivity but in its raw context.

< slackwarepkginfo_test >

The slackware package info test is used to check information associated with a given Slackware package. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a slackwarepkginfo_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 1161: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< slackwarepkginfo_object >

The slackwarepkginfo_object element is used by a slackware package info test to define the object to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A slackware package info object consists of a single name entity that identifies the package being checked.

Extends: oval-def:ObjectType

Child Elements

Table 1162: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityObjectStringType (1..1)	This is the package name to check.
oval-def:filter	n/a (0..unbounded)	

< slackwarepkginfo_state >

The slackwarepkginfo_state element defines the different information that can be used to evaluate the specified package. This includes the version, architecture, and revision. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1163: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	This is the package name to check.
version	Restriction of oval-def:EntityStateAnySimpleType. See schema for details. (0..1)	This is the version number of the package.
architecture	oval-def:EntityStateStringType (0..1)	
revision	oval-def:EntityStateStringType (0..1)	

< systemdunitdependency_test >

The systemdunitdependency_test is used to retrieve information about dependencies of a single systemd unit in the form of a list. This list contains all dependencies, including transitive dependencies. For more information see the output generated by systemctl list-dependencies –plain $unit. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a systemdunitdependency_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 1164: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< systemdunitdependency_object >

The systemdunitdependency_object element is used by a systemdunitdependency_test to define the specific units to check the dependencies of. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 1165: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
unit	oval-def:EntityObjectStringType (1..1)	The unit entity refers to the full systemd unit name, which has a form of "$name.$type". For example "cupsd.service". This name is usually also the filename of the unit configuration file located in the /etc/systemd/ and /usr/lib/systemd/ directories.
oval-def:filter	n/a (0..unbounded)	

< systemdunitdependency_state >

The systemdunitdependency_state element holds dependencies of a specific systemd unit. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1166: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
unit	oval-def:EntityStateStringType (0..1)	The unit entity refers to the full systemd unit name, which has a form of "$name.$type". For example "cupsd.service". This name is usually also the filename of the unit configuration file located in the /etc/systemd/ and /usr/lib/systemd/ directories.
dependency	oval-def:EntityStateStringType (0..1)	The dependency entity refers to the name of a unit that was confirmed to be a dependency of the given unit.

< systemdunitproperty_test >

The systemdunitproperty_test is used to retrieve information about systemd units in form of properties. For more information see the output generated by systemctl show $unit. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a systemdunitproperty_object and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 1167: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< systemdunitproperty_object >

The systemdunitproperty_object element is used by a systemdunitproperty_test to define the specific unit and property combination to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 1168: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
unit	oval-def:EntityObjectStringType (1..1)	The unit entity refers to the full systemd unit name, which has a form of "$name.$type". For example "cupsd.service". This name is usually also the filename of the unit configuration file located in the /etc/systemd/ and /usr/lib/systemd/ directories.
property	oval-def:EntityObjectStringType (1..1)	The property entity refers to the systemd unit property that we are interested in.
oval-def:filter	n/a (0..unbounded)	

< systemdunitproperty_state >

The systemdunitproperty_state element holds information about properties of a specific systemd unit. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1169: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
unit	oval-def:EntityStateStringType (0..1)	The unit entity refers to the full systemd unit name, which has a form of "$name.$type". For example "cupsd.service". This name is usually also the filename of the unit configuration file located in the /etc/systemd/ and /usr/lib/systemd/ directories.
property	oval-def:EntityStateStringType (0..1)	The name of the property associated with a systemd unit.
value	oval-def:EntityStateAnySimpleType (0..1)	The value of the property associated with a systemd unit.

== FileBehaviors ==

The FileBehaviors complex type defines a number of behaviors that allow a more detailed definition of a set of files or file related items to collect. 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.

It is important to note that 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. This is because the 'filepath' element represents an absolute path to a particular file and it is not possible to recurse over a file.

Attributes

Table 1170: Attributes

Attribute	Type	Desc.
max_depth	Restriction of xsd:integer (optional *default*='-1')	'max_depth' defines the maximum depth of recursion to perform when a recurse_direction is specified. A value of '0' is equivalent to no recursion, '1' means to step only one directory level up/down, and so on. The default value is '-1' meaning no limitation. For a 'max_depth' of -1 or any value of 1 or more the starting directory must be considered in the recursive search.

Note that the default recurse_direction behavior is 'none' so even though max_depth specifies no limitation by default, the recurse_direction behavior turns recursion off. Note that this behavior only applies with the equality operation on the path entity.

- - recurse
  - Restriction of xsd:string (optional *default*='symlinks and directories') ('directories', 'symlinks', 'symlinks and directories')
  - 'recurse' defines how to recurse into the path entity, in other words what to follow during recursion. Options include symlinks, directories, or both. Note that a max-depth other than 0 has to be specified for recursion to take place and for this attribute to mean anything. Also note that this behavior does not apply

to Windows systems since they do not support symbolic links. On Windows systems the 'recurse' behavior is always equivalent to directories.

Note that this behavior only applies with the equality operation on the path entity.

- - recurse_direction
  - Restriction of xsd:string (optional *default*='none') ('none', 'up', 'down')
  - 'recurse_direction' defines the direction to recurse, either 'up' to parent directories, or 'down' into child directories. The default value is 'none' for no recursion.

Note that this behavior only applies with the equality operation on the path entity.

- - recurse_file_system
  - Restriction of xsd:string (optional *default*='all') ('all', 'local', 'defined')
  - 'recurse_file_system' defines the file system limitation of any searching and applies to all operations as specified on the path or filepath entity. The value of 'local' limits the search scope to local file systems (as opposed to file systems mounted from an external system). The value of 'defined' keeps any recursion within the file system that the file_object (path+filename or filepath) has specified. For example, if the path specified was "/", you would search only the filesystem mounted there, not other filesystems mounted to descendant paths. The value of 'defined' only applies when an equality operation is used for searching because the path or filepath entity must explicitly define a file system. The default value is 'all' meaning to search all available file systems for data collection.

Note that 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. Searching 'all' file systems may have performance implications.

== EntityStateRpmVerifyResultType ==

The EntityStateRpmVerifyResultType complex type restricts a string value to the set of possible outcomes of checking an attribute of a file included in an RPM against the actual value of that attribute in the RPM database. The empty string is also allowed to support the empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 1171: Enumeration Values

Value	Description
pass	‘pass’ indicates that the test passed and is equivalent to the ‘.’ value reported by the rpm -V command.
fail	‘fail’ indicates that the test failed and is equivalent to a bold charcter in the test result string reported by the rpm -V command.
not performed	‘not performed’ indicates that the test could not be performed and is equivalent to the ‘?’ value reported by the rpm -V command.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateProtocolType ==

The EntityStateProtocolType complex type restricts a string value to the set of physical layer protocols used by AF_PACKET sockets. The empty string is also allowed to support the empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 1172: Enumeration Values

Value	Description
ETH_P_LOOP	Ethernet loopback packet.
ETH_P_PUP	Xerox PUP packet.
ETH_P_PUPAT	Xerox PUP Address Transport packet.
ETH_P_IP	Internet protocol packet.
ETH_P_X25	CCITT X.25 packet.

Continued on next page

Table 1172 – continued from previous page

Value	Description
ETH_P_ARP	Address resolution packet.
ETH_P_BPQ	G8BPQ AX.25 ethernet packet.
ETH_P_IEEEPUP	Xerox IEEE802.3 PUP packet.
ETH_P_IEEEPUPAT	Xerox IEEE802.3 PUP address transport packet.
ETH_P_DEC	DEC assigned protocol.
ETH_P_DNA_DL	DEC DNA Dump/Load.
ETH_P_DNA_RC	DEC DNA Remote Console.
ETH_P_DNA_RT	DEC DNA Routing.
ETH_P_LAT	DEC LAT.
ETH_P_DIAG	DEC Diagnostics.
ETH_P_CUST	DEC Customer use.
ETH_P_SCA	DEC Systems Comms Arch.
ETH_P_RARP	Reverse address resolution packet.
ETH_P_ATALK	Appletalk DDP.
ETH_P_AARP	Appletalk AARP.
ETH_P_8021Q	802.1Q VLAN Extended Header.

Continued on next page

Table 1172 – continued from previous page

Value	Description
ETH_P_IPX	IPX over DIX.
ETH_P_IPV6	IPv6 over bluebook.
ETH_P_SLOW	Slow Protocol. See 802.3ad 43B.
ETH_P_WCCP	Web-cache coordination protocol.
ETH_P_PPP_DISC	PPPoE discovery messages.
ETH_P_PPP_SES	PPPoE session messages.
ETH_P_MPLS_UC	MPLS Unicast traffic.
ETH_P_MPLS_MC	MPLS Multicast traffic.
ETH_P_ATMMPOA	MultiProtocol Over ATM.
ETH_P_ATMFATE	Frame-based ATM Transport over Ethernet.
ETH_P_AOE	ATA over Ethernet.
ETH_P_TIPC	TIPC.
ETH_P_802_3	Dummy type for 802.3 frames.
ETH_P_AX25	Dummy protocol id for AX.25.
ETH_P_ALL	Every packet.
ETH_P_802_2	802.2 frames.

Continued on next page

Table 1172 – continued from previous page

Value	Description
ETH_P_SNAP	Internal only.
ETH_P_DDCMP	DEC DDCMP: Internal only
ETH_P_WAN_PPP	Dummy type for WAN PPP frames.
ETH_P_PPP_MP	Dummy type for PPP MP frames.
ETH_P_PPPTALK	Dummy type for Atalk over PPP.
ETH_P_LOCALTALK	Localtalk pseudo type.
ETH_P_TR_802_2	802.2 frames.
ETH_P_MOBITEX	Mobitex.
ETH_P_CONTROL	Card specific control frames.
ETH_P_IRDA	Linux-IrDA.
ETH_P_ECONET	Acorn Econet.
ETH_P_HDLC	HDLC frames.
ETH_P_ARCNET	1A for ArcNet.
	The empty string value is permitted here to allow for empty elements associated with variable references.

Open Vulnerability and Assessment Language: Linux System Characteristics

- Schema: Linux System Characteristics
- Version: 5.11.1:1.2
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the Linux specific system characteristic items found in Open Vulnerability and Assessment Language (OVAL). Each item is an extension of the standard item element defined in the Core System Characteristic Schema. Through extension, each item inherits a set of elements and attributes that are shared amongst all OVAL Items. Each item is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core System Characteristic Schema is not outlined here.

The OVAL Schema is maintained by the 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.cisecurity.org.

Item Listing

- *< apparmorstatus_item >*
- *< dpkginfo_item >*
- *< iflisteners_item >*
- *< inetlisteningserver_item > (Deprecated)*
- *< partition_item >*
- *< rpminfo_item >*
- *< rpmverify_item > (Deprecated)*
- *< rpmverifyfile_item >*
- *< rpmverifypackage_item >*
- *< selinuxboolean_item >*
- *< selinuxsecuritycontext_item >*
- *< slackwarepkginfo_item >*
- *< systemdunitdependency_item >*
- *< systemdunitproperty_item >*

< apparmorstatus_item >

The AppArmor Status Item displays various information about the current AppArmor policy. This item maps the counts of profiles and processes as per the results of the “apparmor_status” or “aa-status” command. Each item extends the standard ItemType as defined in the oval-system-characteristics-schema and one should refer to the ItemType description for more information.

Extends: oval-sc:ItemType

Child Elements

Table 1173: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
loaded_profiles_count	oval-sc:EntityItemIntType (0..1)	Displays the number of loaded profiles
enforce_mode_profiles_count	oval-sc:EntityItemIntType (0..1)	Displays the number of profiles in enforce mode
complain_mode_profiles_count	oval-sc:EntityItemIntType (0..1)	Displays the number of profiles in complain mode
processes_with_profiles_count	oval-sc:EntityItemIntType (0..1)	Displays the number of processes which have profiles defined
enforce_mode_processes_count	oval-sc:EntityItemIntType (0..1)	Displays the number of processes in enforce mode
complain_mode_processes_count	oval-sc:EntityItemIntType (0..1)	Displays the number of processes in complain mode
unconfined_processes_with_profiles_count	oval-sc:EntityItemIntType (0..1)	Displays the number of processes which are unconfined but have a profile defined

---

< dpkginfo_item >

This item stores DPKG package info.

Extends: oval-sc:ItemType

Child Elements

Table 1174: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	This is the pakage name to check.
arch	oval-sc:EntityItemStringType (0..1)	This is the architecture for which the DPKG was built, like : i386, ppc, sparc, noarch.
epoch	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	This is the epoch number of the DPKG. For a null epoch (or '(none)' as returned by dpkg) the string '(none)' should be used.
release	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	This is the release number of the build.
version	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	This is the version number of the build, changed by the vendor/builder.
evr	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	This type represents the epoch, upstream_version, and debian_revision fields, for a Debian package, as a single version string. It has the form "EPOCH:UPSTREAM_VERSION-DEBIAN_REVISION". Note that a null epoch (or '(none)' as returned by dpkg) is equivalent to '0' and would hence have the form 0:UPSTREAM_VERSION-DEBIAN_REVISION.

< iflisteners_item >

An iflisteners_item stores the results of checking for applications that are bound to an interface on the system. Only applications that are bound to an ethernet interface should be collected.

Extends: oval-sc:ItemType

Child Elements

Table 1175: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
interface_name	oval-sc:EntityItemStringType (0..1)	This is the name of the interface (eth0, eth1, fw0, etc.).
protocol	linux-sc:EntityItemProtocolType (0..1)	This is the physical layer protocol used by the AF_PACKET socket.
hw_address	oval-sc:EntityItemStringType (0..1)	This is the hardware address associated with the interface.
program_name	oval-sc:EntityItemStringType (0..1)	This is the name of the communicating program.
pid	oval-sc:EntityItemIntType (0..1)	This is the process ID of the process. The process in question is that of the program communicating on the network.
user_id	oval-sc:EntityItemIntType (0..1)	The numeric user id, or uid, is the third column of each user's entry in /etc/passwd. It represents the owner, and thus privilege level, of the specified program.

< inetlisteningserver_item >

An inet listening server item stores the results of checking for network servers currently active on a system. It holds information pertaining to a specific protocol-address-port combination.

Extends: oval-sc:ItemType

Child Elements

Table 1176: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
protocol	oval-sc:EntityItemStringType (0..1)	This is the transport-layer protocol, in lowercase: tcp or udp.
local_address	oval-sc:EntityItemIPAddressStringType (0..1)	This is the IP address associated with the inet listening server. Note that the IP address can be IPv4 or IPv6.
local_port	oval-sc:EntityItemIntType (0..1)	This is the TCP or UDP port on which the program listens.
local_full_address	oval-sc:EntityItemStringType (0..1)	This is the IP address and network port on which the program listens, equivalent to local_address:local_port. Note that the IP address can be IPv4 or IPv6.
program_name	oval-sc:EntityItemStringType (0..1)	This is the name of the communicating program.
foreign_address	oval-sc:EntityItemIPAddressStringType (0..1)	This is the IP address with which the program is communicating, or with which it will communicate, in the case of a listening server. Note that the IP address can be IPv4 or IPv6.
foreign_port	oval-sc:EntityItemIntType (0..1)	This is the TCP or UDP port to which the program communicates. In the case of a listening program accepting new connections, this value will be 0.
foreign_full_address	oval-sc:EntityItemStringType (0..1)	This is the IP address and network port to which the program is communicating or will accept communications from, equivalent to foreign_address:foreign_port. Note that the IP address can be IPv4 or IPv6.
pid	oval-sc:EntityItemIntType (0..1)	This is the process ID of the process. The process in question is that of the program communicating on the network.
user_id	oval-sc:EntityItemIntType (0..1)	The numeric user id, or uid, is the third column of each user's entry in /etc/passwd. It represents the owner, and thus privilege level, of the specified program.

< partition_item >

The partition_item stores information about a partition on the local system.

Extends: oval-sc:ItemType

Child Elements

Table 1177: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
mount_point	oval-sc:EntityItemStringType (0..1)	The mount_point element contains a string that represents the mount point of a partition on the local system.
device	oval-sc:EntityItemStringType (0..1)	The device element contains a string that represents the name of the device.
uuid	oval-sc:EntityItemStringType (0..1)	The uuid element contains a string that represents the universally unique identifier associated with a partition.
fs_type	oval-sc:EntityItemStringType (0..1)	The fs_type element contains a string that represents the type of filesystem on a partition.
mount_options	oval-sc:EntityItemStringType (0..unbounded)	The mount_options element contains a string that represents a mount option associated with a partition on the local system.Implementation note: not all mount options are visible in /etc/mtab or /proc/mounts. A complete source of additional mount options is the f_flag field of 'struct statvfs'. See statvfs(2). /etc/fstab may have additional mount options, but it need not contain all mounted filesystems, so it MUST NOT be relied upon. Implementers MUST be sure to get all mount options in some way.
total_space	oval-sc:EntityItemIntType (0..1)	The total_space element contains an integer that represents the total number of physical blocks on a partition.
space_used	oval-sc:EntityItemIntType (0..1)	The space_used element contains an integer that represents the number of physical blocks used on a partition.
space_left	oval-sc:EntityItemIntType (0..1)	The space_left element contains an integer that represents the number of physical blocks left on a partition available to be used by privileged users.
space_left_for_unprivileged_users	oval-sc:EntityItemIntType (0..1)	The space_left_for_unprivileged_users element contains an integer that represents the number of physical blocks remaining on a partition that are available to be used by unprivileged users.
block_size	oval-sc:EntityItemIntType (0..1)	The block_size element contains an integer representing the actual byte size of each physical block on the partition's block device. This is the same block size used to compute the total_space, space_used, and space_left.

< rpminfo_item >

This item stores rpm info.

Extends: oval-sc:ItemType

Child Elements

Table 1178: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	This is the pakage name to check.
arch	oval-sc:EntityItemStringType (0..1)	This is the architecture for which the RPM was built, like : i386, ppc, sparc, noarch. In the case of an apache rpm named httpd-2.0.40-21.11.4.i686.rpm, this value would be i686.
epoch	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	This is the epoch number of the RPM, this is used as a kludge for version-release comparisons where the vendor has done some kind of re-numbering or version forking. For a null epoch (or '(none)' as returned by rpm) the string '(none)' should be used. This number is not revealed by a normal query of the RPM's information – you must use a formatted rpm query command to gather this data from the command line, like so. For an already-installed RPM: rpm -q –qf '%{EPOCH}n' installed_rpm For an RPM file that has not been installed: rpm -qp –qf '%{EPOCH}n' rpm_file
release	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	This is the release number of the build.
version	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	This is the version number of the build, changed by the vendor/builder. In the case of an apache rpm named httpd-2.0.40-21.11.4.i686.rpm, this value would be 2.0.40.
evr	oval-sc:EntityItemEVRStringType (0..1)	This represents the epoch, version, and release fields as a single version string. It has the form "EPOCH:VERSION-RELEASE". Note that a null epoch (or '(none)' as returned by rpm) is equivalent to '0' and would hence have the form 0:VERSION-RELEASE.
signature_keyid	oval-sc:EntityItemStringType (0..1)	This field contains the PGP key ID that the RPM issuer (generally the original operating system vendor) uses to sign the key. PGP is used to verify the authenticity and integrity of the RPM being considered. Software packages and patches are signed cryptographically to allow administrators to allay concerns that the distribution mechanism has been compromised, whether that mechanism is web site, FTP server, or even a mirror controlled by a hostile party. OVAL uses this field most of all to confirm that the package installed on the system is that shipped by the vendor, since comparing package version numbers against patch announcements is only programmatically valid if the installed package is known to contain the patched code.
extended_name	oval-sc:EntityItemStringType (0..1)	This represents the name, epoch, version, release, and architecture fields as a single version string. It has the form "NAME-EPOCH:VERSION-RELEASE.ARCHITECTURE". Note that a null epoch (or '(none)' as returned by rpm) is equivalent to '0' and would hence have the form NAME-0:VERSION-RELEASE.ARCHITECTURE. The 'gpg-pubkey' virtual package on RedHat and CentOS should use the string '(none)' for the architecture to construct the extended_name.
filepath	oval-sc:EntityItemStringType	This field contains the absolute path of a file or directory included in the rpm.

< rpmverify_item > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.10
- Reason: Replaced by the rpmverifyfile_item and rpmverifypackage_item. The rpmverify_item was split into two items to distinguish between the verification of the files in an rpm and the verification of an rpm as a whole. By making this distinction, content authoring is simplified and information is no longer duplicated across items. See the rpmverifyfile_item and rpmverifypackage_item.
- Comment: This state has been deprecated and will be removed in version 6.0 of the language.

This item stores rpm verification results similar to what is produced by the rpm -V command.

Extends: oval-sc:ItemType

Child Elements

Table 1179: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	This is the package name to check.
filepath	oval-sc:EntityItemStringType (0..1)	The filepath element specifies the absolute path for a file or directory in the specified package.
size_differs	linux-sc:EntityItemRpmVerifyResultType (0..1)	The size_differs entity aligns with the first character ('S' flag) in the character string in the output generated by running rpm –V on a specific file.
mode_differs	linux-sc:EntityItemRpmVerifyResultType (0..1)	The mode_differs entity aligns with the second character ('M' flag) in the character string in the output generated by running rpm –V on a specific file.
md5_differs	linux-sc:EntityItemRpmVerifyResultType (0..1)	The md5_differs entity aligns with the third character ('5' flag) in the character string in the output generated by running rpm –V on a specific file.
device_differs	linux-sc:EntityItemRpmVerifyResultType (0..1)	The device_differs entity aligns with the fourth character ('D' flag) in the character string in the output generated by running rpm –V on a specific file.
link_mismatch	linux-sc:EntityItemRpmVerifyResultType (0..1)	The link_mismatch entity aligns with the fifth character ('L' flag) in the character string in the output generated by running rpm –V on a specific file.
ownership_differs	linux-sc:EntityItemRpmVerifyResultType (0..1)	The ownership_differs entity aligns with the sixth character ('U' flag) in the character string in the output generated by running rpm –V on a specific file.
group_differs	linux-sc:EntityItemRpmVerifyResultType (0..1)	The group_differs entity aligns with the seventh character ('U' flag) in the character string in the output generated by running rpm –V on a specific file.
mtime_differs	linux-sc:EntityItemRpmVerifyResultType (0..1)	The mtime_differs entity aligns with the eighth character ('T' flag) in the character string in the output generated by running rpm –V on a specific file.
capabilities_differ	linux-sc:EntityItemRpmVerifyResultType (0..1)	The size_differs entity aligns with the ninth character ('P' flag) in the character string in the output generated by running rpm –V on a specific file.
configuration_file	oval-sc:EntityItemBoolType (0..1)	The configuration_file entity represents the configuration file attribute marker that may be present on a file.
documentation_file	oval-sc:EntityItemBoolType (0..1)	The documentation_file entity represents the documenation file attribute marker that may be present on a file.
ghost_file	oval-sc:EntityItemBoolType (0..1)	The ghost_file entity represents the ghost file attribute marker that may be present on a file.
license_file	oval-sc:EntityItemBoolType (0..1)	The license_file entity represents the license file attribute marker that may be present on a file.
readme_file	oval-sc:EntityItemBoolType (0..1)	The readme_file entity represents the readme file attribute marker that may be present on a file.

< rpmverifyfile_item >

This item stores the verification results of the individual files in an rpm similar to what is produced by the rpm -V command.

Extends: oval-sc:ItemType

Child Elements

Table 1180: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	This is the package name to check.
epoch	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	This is the epoch number of the RPM, this is used as a kludge for version-release comparisons where the vendor has done some kind of re-numbering or version forking. For a null epoch (or '(none)' as returned by rpm) the string '(none)' should be used.. This number is not revealed by a normal query of the RPM's information – you must use a formatted rpm query command to gather this data from the command line, like so. For an already-installed RPM: rpm -q –qf '%{EPOCH}n' installed_rpm For an RPM file that has not been installed: rpm -qp –qf '%{EPOCH}n' rpm_file
version	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	This is the version number of the build. In the case of an apache rpm named httpd-2.0.40-21.11.4.i686.rpm, this value would be 2.0.40.
release	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	This is the release number of the build, changed by the vendor/builder.
arch	oval-sc:EntityItemStringType (0..1)	This is the architecture for which the RPM was built, like : i386, ppc, sparc, noarch. In the case of an apache rpm named httpd-2.0.40-21.11.4.i686.rpm, this value would be i686.
filepath	oval-sc:EntityItemStringType (0..1)	The filepath element specifies the absolute path for a file or directory in the specified package.
extended_name	oval-sc:EntityItemStringType (0..1)	This represents the name, epoch, version, release, and architecture fields as a single version string. It has the form "NAME-EPOCH:VERSION-RELEASE.ARCHITECTURE". Note that a null epoch (or '(none)' as returned by rpm) is equivalent to '0' and would hence have the form NAME-0:VERSION-RELEASE.ARCHITECTURE.
size_differs	linux-sc:EntityItemRpmVerifyResultType (0..1)	The size_differs entity aligns with the first character ('S' flag) in the character string in the output generated by running rpm –V on a specific file.
mode_differs	linux-sc:EntityItemRpmVerifyResultType (0..1)	The mode_differs entity aligns with the second character ('M' flag) in the character string in the output generated by running rpm –V on a specific file.
md5_differs (Deprecated)	linux-sc:EntityItemRpmVerifyResultType (0..1)	The md5_differs entity aligns with the third character ('5' flag) in the character string in the output generated by running rpm –V on a specific file.
filedigest_differs	linux-sc:EntityItemRpmVerifyResultType (0..1)	The filedigest_differs entity aligns with the third character ('5' flag) in the character string in the output generated by running rpm –V on a specific file. This replaces the md5_differs entity due to naming changes for verification and reporting options.
device_differs	linux-sc:EntityItemRpmVerifyResultType (0..1)	The device_differs entity aligns with the fourth character ('D' flag) in the character string in the output generated by running rpm –V on a specific file.
link_mismatch	linux-sc:EntityItemRpmVerifyResultType (0..1)	The link_mismatch entity aligns with the fifth character ('L' flag) in the character string in the output generated by running rpm –V on a specific file.
own-	linux-	The ownership_differs entity aligns with the sixth character ('U' flag) in the character string

< rpmverifypackage_item >

This item stores the rpm verification results of an rpm similar to what is produced by the rpm -V command.

Extends: oval-sc:ItemType

Child Elements

Table 1181: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	This is the package name to check.
epoch	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	This is the epoch number of the RPM, this is used as a kludge for version-release comparisons where the vendor has done some kind of re-numbering or version forking. For a null epoch (or '(none)' as returned by rpm) the string '(none)' should be used.. This number is not revealed by a normal query of the RPM's information – you must use a formatted rpm query command to gather this data from the command line, like so. For an already-installed RPM: rpm -q –qf '%{EPOCH}n' installed_rpm For an RPM file that has not been installed: rpm -qp –qf '%{EPOCH}n' rpm_file
version	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	This is the version number of the build. In the case of an apache rpm named httpd-2.0.40-21.11.4.i686.rpm, this value would be 2.0.40.
release	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	This is the release number of the build, changed by the vendor/builder.
arch	oval-sc:EntityItemStringType (0..1)	This is the architecture for which the RPM was built, like : i386, ppc, sparc, noarch. In the case of an apache rpm named httpd-2.0.40-21.11.4.i686.rpm, this value would be i686.
extended_name	oval-sc:EntityItemStringType (0..1)	This represents the name, epoch, version, release, and architecture fields as a single version string. It has the form "NAME-EPOCH:VERSION-RELEASE.ARCHITECTURE". Note that a null epoch (or '(none)' as returned by rpm) is equivalent to '0' and would hence have the form NAME-0:VERSION-RELEASE.ARCHITECTURE.
dependency_check_passed	oval-sc:EntityItemBoolType (0..1)	The dependency_check_passed entity indicates whether or not the dependency check passed. If the dependency check is not performed, due to the 'nodeps' behavior, this entity must not be collected.
digest_check_passed (Deprecated)	oval-sc:EntityItemBoolType (0..1)	The digest_check_passed entity indicates whether or not the verification of the package or header digests passed. If the digest check is not performed, due to the 'nodigest' behavior, this entity must not be collected.
verification_script_successful	oval-sc:EntityItemBoolType (0..1)	The verification_script_successful entity indicates whether or not the verification script executed successfully. If the verification script is not executed, due to the 'noscripts' behavior, this entity must not be collected.
signature_check_passed (Deprecated)	oval-sc:EntityItemBoolType (0..1)	The signature_check_passed entity indicates whether or not the verification of the package or header signatures passed. If the signature check is not performed, due to the 'nosignature' behavior, this entity must not be collected.

< selinuxboolean_item >

This item describes the current and pending status of a SELinux boolean. It extends the standard ItemType as defined in the oval-system-characteristics schema and one should refer to the ItemType description for more information.

Extends: oval-sc:ItemType

Child Elements

Table 1182: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	The name of the SELinux boolean.
current_status	oval-sc:EntityItemBoolType (0..1)	The current_status entity indicates current state of the specified SELinux boolean.
pending_status	oval-sc:EntityItemBoolType (0..1)	The pending_status entity indicates the pending state of the specified SELinux boolean.

< selinuxsecuritycontext_item >

This item describes the SELinux security context of a file or process on the local system. This item follows the SELinux security context structure: user:role:type:low_sensitivity[:low_category]- high_sensitivity [:high_category]. It extends the standard ItemType as defined in the oval-system-characteristics schema and one should refer to the ItemType description for more information.

Extends: oval-sc:ItemType

Child Elements

Table 1183: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
filepath	oval-sc:EntityItemStringType (0..1)	The filepath element specifies the absolute path for a file on the machine. A directory cannot be specified as a filepath.
path	oval-sc:EntityItemStringType (0..1)	The path element specifies the directory component of the absolute path to a file on the machine.
filename	oval-sc:EntityItemStringType (0..1)	The name of the file. If the xsi:nil attribute is set to true, then the item being represented is the higher directory represented by the path entity.
pid	oval-sc:EntityItemIntType (0..1)	This is the process ID of the process.
user	oval-sc:EntityItemStringType (0..1)	The user element specifies the SELinux user that either created the file or started the process.
role	oval-sc:EntityItemStringType (0..1)	The role element specifies the types that a process may transition to (domain transitions). Note that this entity is not relevant for files and will always have a value of object_r.
type	oval-sc:EntityItemStringType (0..1)	The type element specifies the domain in which the file is accessible or the domain in which a process executes.
low_sensitivity	oval-sc:EntityItemStringType (0..1)	The low_sensitivity element specifies the current sensitivity of a file or process.
low_category	oval-sc:EntityItemStringType (0..1)	The low_category element specifies the set of categories associated with the low sensitivity.
high_sensitivity	oval-sc:EntityItemStringType (0..1)	The high_sensitivity element specifies the maximum range for a file or the clearance for a process.
high_category	oval-sc:EntityItemStringType (0..1)	The high_category element specifies the set of categories associated with the high sensitivity.
rawlow_sensitivity	oval-sc:EntityItemStringType (0..1)	The rawlow_sensitivity element specifies the current sensitivity of a file or process but in its raw context.
rawlow_category	oval-sc:EntityItemStringType (0..1)	The rawlow_category element specifies the set of categories associated with the low sensitivity but in its raw context.
rawhigh_sensitivity	oval-sc:EntityItemStringType (0..1)	The rawhigh_sensitivity element specifies the maximum range for a file or the clearance for a process but in its raw context.
rawhigh_category	oval-sc:EntityItemStringType (0..1)	The rawhigh_category element specifies the set of categories associated with the high sensitivity but in its raw context.

< slackwarepkginfo_item >

This item describes info related to Slackware packages. It extends the standard ItemType as defined in the oval-system-characteristics schema and one should refer to the ItemType description for more information.

Extends: oval-sc:ItemType

Child Elements

Table 1184: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	This is the pakage name to check.
version	Restriction of oval-sc:EntityItemAnySimpleType. See schema for details. (0..1)	This is the version number of the pakage.
architecture	oval-sc:EntityItemStringType (0..1)	This is the architecture the package is designed for.
revision	oval-sc:EntityItemStringType (0..1)	This is the revision of the package.

< systemdunitdependency_item >

This item stores the dependencies of the systemd unit. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-sc:ItemType

Child Elements

Table 1185: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
unit	oval-sc:EntityItemStringType (0..1)	The unit entity refers to the full systemd unit name, which has a form of “$name.$type”. For example “cupsd.service”. This name is usually also the filename of the unit configuration file located in the /etc/systemd/ and /usr/lib/systemd/ directories.
dependency	oval-sc:EntityItemStringType (0..unbounded)	The dependency entity refers to the name of a unit that was confirmed to be a dependency of the given unit.

< systemdunitproperty_item >

This item stores the properties and values of a systemd unit.

Extends: oval-sc:ItemType

Child Elements

Table 1186: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
unit	oval-sc:EntityItemStringType (0..1)	The unit entity refers to the full systemd unit name, which has a form of "$name.$type". For example "cupsd.service". This name is usually also the filename of the unit configuration file located in the /etc/systemd/ and /usr/lib/systemd/ directories.
property	oval-sc:EntityItemStringType (0..1)	The name of the property associated with a systemd unit.
value	oval-sc:EntityItemAnySimpleType (0..unbounded)	The value of the property associated with a systemd unit. Exactly one value shall be used for all property types except dbus arrays - each array element shall be represented by one value.

== EntityItemRpmVerifyResultType ==

The EntityItemRpmVerifyResultType complex type restricts a string value to the set of possible outcomes of checking an attribute of a file included in an RPM against the actual value of that attribute in the RPM database. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 1187: Enumeration Values

Value	Description
pass	'pass' indicates that the test passed and is equivalent to the '.' value reported by the rpm -V command.
fail	'fail' indicates that the test failed and is equivalent to a bold charcter in the test result string reported by the rpm -V command.
not performed	'not performed' indicates that the test could not be performed and is equivalent to the '?' value reported by the rpm -V command.
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemProtocolType ==

The EntityStateProtocolType complex type restricts a string value to the set of physical layer protocols used by AF_PACKET sockets. The empty string is also allowed to support the empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-sc:EntityItemStringType

Table 1188: Enumeration Values

Value	Description
ETH_P_LOOP	Ethernet loopback packet.
ETH_P_PUP	Xerox PUP packet.
ETH_P_PUPAT	Xerox PUP Address Transport packet.
ETH_P_IP	Internet protocol packet.
ETH_P_X25	CCITT X.25 packet.
ETH_P_ARP	Address resolution packet.
ETH_P_BPQ	G8BPQ AX.25 ethernet packet.
ETH_P_IEEEPUP	Xerox IEEE802.3 PUP packet.
ETH_P_IEEEPUPAT	Xerox IEEE802.3 PUP address transport packet.
ETH_P_DEC	DEC assigned protocol.
ETH_P_DNA_DL	DEC DNA Dump/Load.
ETH_P_DNA_RC	DEC DNA Remote Console.
ETH_P_DNA_RT	DEC DNA Routing.

Continued on next page

Table 1188 – continued from previous page

Value	Description
ETH_P_LAT	DEC LAT.
ETH_P_DIAG	DEC Diagnostics.
ETH_P_CUST	DEC Customer use.
ETH_P_SCA	DEC Systems Comms Arch.
ETH_P_RARP	Reverse address resolution packet.
ETH_P_ATALK	Appletalk DDP.
ETH_P_AARP	Appletalk AARP.
ETH_P_8021Q	802.1Q VLAN Extended Header.
ETH_P_IPX	IPX over DIX.
ETH_P_IPV6	IPv6 over bluebook.
ETH_P_SLOW	Slow Protocol. See 802.3ad 43B.
ETH_P_WCCP	Web-cache coordination protocol.
ETH_P_PPP_DISC	PPPoE discovery messages.
ETH_P_PPP_SES	PPPoE session messages.
ETH_P_MPLS_UC	MPLS Unicast traffic.
ETH_P_MPLS_MC	MPLS Multicast traffic.

Continued on next page

Table 1188 – continued from previous page

Value	Description
ETH_P_ATMMPOA	MultiProtocol Over ATM.
ETH_P_ATMFATE	Frame-based ATM Transport over Ethernet.
ETH_P_AOE	ATA over Ethernet.
ETH_P_TIPC	TIPC.
ETH_P_802_3	Dummy type for 802.3 frames.
ETH_P_AX25	Dummy protocol id for AX.25.
ETH_P_ALL	Every packet.
ETH_P_802_2	802.2 frames.
ETH_P_SNAP	Internal only.
ETH_P_DDCMP	DEC DDCMP: Internal only
ETH_P_WAN_PPP	Dummy type for WAN PPP frames.
ETH_P_PPP_MP	Dummy type for PPP MP frames.
ETH_P_PPPTALK	Dummy type for Atalk over PPP.
ETH_P_LOCALTALK	Localtalk pseudo type.
ETH_P_TR_802_2	802.2 frames.
ETH_P_MOBITEX	Mobitex.

Continued on next page

Table 1188 – continued from previous page

Value	Description
ETH_P_CONTROL	Card specific control frames.
ETH_P_IRDA	Linux-IrDA.
ETH_P_ECONET	Acorn Econet.
ETH_P_HDLC	HDLC frames.
ETH_P_ARCNET	1A for ArcNet.
	The empty string value is permitted here to allow for detailed error reporting.

Open Vulnerability and Assessment Language: Solaris Definition

- Schema: Solaris Definition
- Version: 5.11.1:1.1
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the Solaris specific tests found in Open Vulnerability and Assessment Language (OVAL). Each test is an extension of the standard test element defined in the Core Definition Schema. Through extension, each test inherits a set of elements and attributes that are shared amongst all OVAL tests. Each test is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core Definition Schema is not outlined here.

The OVAL Schema is maintained by the 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.cisecurity.org.

Test Listing

- *< facet_test >*
- *< image_test >*
- *< isainfo_test >*
- *< ndd_test >*
- *< package_test >*
- *< package511_test >*
- *< packageavoidlist_test >*

- *< packagecheck_test >*
- *< packagefreezelist_test >*
- *< packagepublisher_test >*
- *< patch54_test >*
- *< patch_test > (Deprecated)* (Deprecated)
- *< smf_test >*
- *< smfproperty_test >*
- *< variant_test >*
- *< virtualizationinfo_test >*

< facet_test >

The facet_test is used to check the facets associated with the specified Image Packaging System image. Facets are properties that control whether or not optional components from a package are installed on a system. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an facet_object and the optional state elements reference a facet_state and specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 1189: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< facet_object >

The facet_object element is used by a facet test to define the image facet items to be evaluated based on the specified states. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 1190: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
path	oval-def:EntityObjectStringType (1..1)	The path to the Solaris IPS image.
name	oval-def:EntityObjectStringType (1..1)	The name of the facet property associated with an IPS image.
oval-def:filter	n/a (0..unbounded)	

< facet_state >

The facet_state specifies the various facet properties associated with an IPS image.

Extends: oval-def:StateType

Child Elements

Table 1191: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
path	oval-def:EntityStateStringType (0..1)	Specifies the path to the Solaris IPS image.
name	oval-def:EntityStateStringType (0..1)	Specifies the name of the facet property associated with an IPS image.
value	oval-def:EntityStateBoolType (0..1)	Specifies the value of the facet property associated with an IPS image.

< image_test >

The image_test provides support for checking the metadata of IPS images on Solaris systems. The test extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a image_object and the optional state elements reference image_states that specify the metadata to check about a set of images.

Extends: oval-def:TestType

Child Elements

Table 1192: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< image_object >

The image_object element is used by a image_test to identify the set of images to check on a system. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 1193: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
path	oval-def:EntityObjectStringType (1..1)	The path to the Solaris IPS image.
name	oval-def:EntityStateStringType (1..1)	The name of the property associated with the Solaris IPS image.
oval-def:filter	n/a (0..unbounded)	

< image_state >

The image_state element defines the different system state information that can be used to check the metadata associated with the specified IPS image on a Solaris system.

Extends: oval-def:StateType

Child Elements

Table 1194: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
path	oval-def:EntityStateStringType (0..1)	The path to the Solaris IPS image.
name	oval-def:EntityStateStringType (0..1)	The name of the property associated with the Solaris IPS image.
value	oval-def:EntityStateAnySimpleType (0..1)	The value of a property that is associated with a Solaris IPS image.

< isainfo_test >

The isainfo test reveals information about the instruction set architectures. This information can be retrieved by the isainfo command. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an isainfo_object and the optional state element specifies the metadata to check.

The isainfo_test was originally developed by Robert L. Hollis at ThreatGuard, Inc. Many thanks for their support of the OVAL project.

Extends: oval-def:TestType

Child Elements

Table 1195: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< isainfo_object >

The isainfo_object element is used by an isainfo test to define those objects to evaluated based on a specified state. There is actually only one object relating to isainfo and this is the system as a whole. Therefore, there are no child entities defined. Any OVAL Test written to check isainfo will reference the same isainfo_object which is basically an empty object element.

Extends: oval-def:ObjectType

< isainfo_state >

The isainfo_state element defines the information about the instruction set architectures. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1196: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
bits	oval-def:EntityStateIntType (0..1)	This is the number of bits in the address space of the native instruction set (isainfo -b).
kernel_isa	oval-def:EntityStateStringType (0..1)	This is the name of the instruction set used by kernel components (isainfo -k).
application_isa	oval-def:EntityStateStringType (0..1)	This is the name of the instruction set used by portable applications (isainfo -n).

< ndd_test >

From /usr/bin/ndd. See ndd manpage for specific fields

Extends: oval-def:TestType

Child Elements

Table 1197: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< ndd_object >

Extends: oval-def:ObjectType

Child Elements

Table 1198: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
device	oval-def:EntityObjectStringType (1..1)	The name of the device to examine. If multiple instances of this device exist on the system, an item for each instance will be collected.
parameter	oval-def:EntityObjectStringType (1..1)	The name of the parameter, For example, ip_forwarding.
oval-def:filter	n/a (0..unbounded)	

< ndd_state >

Extends: oval-def:StateType

Child Elements

Table 1199: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
device	oval-def:EntityStateStringType (0..1)	The name of the device to examine.
instance	oval-def:EntityStateIntType (0..1)	The instance of the device to examine. Certain devices may have multiple instances on a system. If multiple instances exist, an item for each instance will be collected and will have this entity populated with its respective instance value. If only a single instance exists, this entity will not be collected.
parameter	oval-def:EntityStateStringType (0..1)	The name of the parameter, For example, ip_forwarding.
value	oval-def:EntityStateAnySimpleType (0..1)	The value of the named parameter.

< package_test >

The package test is used to check information associated with different SVR4 packages installed on the system. Image Packaging System (IPS) packages are not supported by this test. The information used by this test is modeled after the /usr/bin/pkginfo command. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an package_object and the optional state element specifies the information to check.

Extends: oval-def:TestType

Child Elements

Table 1200: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< package_object >

The package_object element is used by a package test to define the SVR4 packages to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A package object consists of a single pkginst entity that identifies the package to be used.

Extends: oval-def:ObjectType

Child Elements

Table 1201: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
pkginst	oval-def:EntityObjectStringType (1..1)	The pkginst entity is a string that represents a package designation by its instance. An instance can be the package abbreviation or a specific instance (for example, inst.1 or inst.2).
oval-def:filter	n/a (0..unbounded)	

< package_state >

The package_state element defines the different information associated with SVR4 packages installed on the system. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1202: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
pkginst	oval-def:EntityStateStringType (0..1)	The pkginst entity is a string that represents a package designation by its instance. An instance can be the package abbreviation or a specific instance (for example, inst.1 or inst.2).
name	oval-def:EntityStateStringType (0..1)	The name entity is a text string that specifies a full package name.
category	oval-def:EntityStateStringType (0..1)	The category entity is a string in the form of a comma-separated list of categories under which a package may be displayed. Note that a package must at least belong to the system or application category. Categories are case-insensitive and may contain only alphanumerics. Each category is limited in length to 16 characters.
version	oval-def:EntityStateStringType (0..1)	The version entity is a text string that specifies the current version associated with the software package. The maximum length is 256 ASCII characters and the first character cannot be a left parenthesis. Current Solaris software practice is to assign this parameter monotonically increasing Dewey decimal values of the form: major_revision.minor_revision[.micro_revision] where all the revision fields are integers. The versioning fields can be extended to an arbitrary string of numbers in Dewey-decimal format, if necessary.
vendor	oval-def:EntityStateStringType (0..1)	The vendor entity is a string used to identify the vendor that holds the software copyright (maximum length of 256 ASCII characters).
description	oval-def:EntityStateStringType (0..1)	The description entity is a string that represents a more in-depth description of a package.

< package511_test >

The package511_test provides support for checking the metadata of packages installed using the Solaris Image Packaging System. The test extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a package511_object and the optional state elements reference package511_states that specify the metadata to check about a set of packages.

Extends: oval-def:TestType

Child Elements

Table 1203: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< package511_object >

The package511_object element is used by a package511_test to identify the set of packages to check on a system. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 1204: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
publisher	oval-def:EntityObjectStringType (1..1)	The person, group of persons, or organization that is the source of the package. The publisher should be expressed without leading “pkg:” or “//” components.
name	oval-def:EntityObjectStringType (1..1)	The full hierarchical name of the package which is separated by forward slash characters. The full name should be expressed without leading “pkg:/” or “/” components.
version	oval-def:EntityObjectVersionType (1..1)	The version of the package which consists of the component version, build version, and branch version.
timestamp	oval-def:EntityObjectStringType (1..1)	The timestamp when the package was published in the ISO-8601 basic format (YYYYMMDDTHHMMSSZ).
oval-def:filter	n/a (0..unbounded)	

< package511_state >

The package511_state element defines the different system state information that can be used to check the metadata associated with the specified IPS packages on a Solaris system.

Extends: oval-def:StateType

Child Elements

Table 1205: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
publisher	oval-def:EntityStateStringType (0..1)	The person, group of persons, or organization that is the source of the package. The publisher should be expressed without leading “pkg:” or “//” components.
name	oval-def:EntityStateStringType (0..1)	The full hierarchical name of the package which is separated by forward slash characters. The full name should be expressed without leading “pkg:/” or “/” components.
version	oval-def:EntityStateVersionType (0..1)	The version of the package which consists of the component version, build version, and branch version.
timestamp	oval-def:EntityStateStringType (0..1)	The timestamp when the package was published in the ISO-8601 basic format (YYYYMMDDTHHMMSSZ).
fmri	oval-def:EntityStateStringType (0..1)	The Fault Management Resource Identifier (FMRI) of the package which uniquely identifies the package on the system.
summary	oval-def:EntityStateStringType (0..1)	A summary of what the package provides.
description	oval-def:EntityStateStringType (0..1)	A description of what the package provides.
category	oval-def:EntityStateStringType (0..1)	The category of the package.
updates_available	oval-def:EntityStateBoolType (0..1)	A boolean value indicating whether or not updates are available for this package.

< packageavoidlist_test >

The packageavoidlist_test provides support for checking the metadata of IPS packages that have been flagged as needing to avoid from installation on a Solaris system. The test extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a packageavoidlist_object and the optional state elements reference packageavoidlist_states that specify the metadata to check about a set of packages that have been flagged as to be avoided on a Solaris system.

Extends: oval-def:TestType

Child Elements

Table 1206: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< packageavoidlist_object >

The packageavoidlist_object element is used by a packageavoidlist_test to identify the set of IPS packages that have been flagged as to be avoided from installation on a Solaris system. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

< packageavoidlist_state >

The packageavoidlist_state element defines the different system state information that can be used to evaluate the specified IPS packages that have been flagged as to be avoided from installation on a Solaris system.

Extends: oval-def:StateType

Child Elements

Table 1207: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
fmri	oval-def:EntityStateStringType (0..1)	The Fault Management Resource Identifier (FMRI) of the package which uniquely identifies the package on the system.

< packagecheck_test >

The packagecheck_test is used to verify the integrity of an installed Solaris SVR4 package. Image Packaging System (IPS) packages are not supported by this test. The information used by this test is modeled after the pkgchk command. For more information, see pkgchk(1M). It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a packagecheck_object and the optional packagecheck_state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 1208: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< packagecheck_object >

The packagecheck_object element is used by a packagecheck_test to define the SVR4 packages to be verified. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 1209: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	sol-def:PackageCheckBehaviors (0..1)	
pkginst	oval-def:EntityObjectStringType (1..1)	The pkginst entity is a string that represents a package designation by its instance. An instance can be the package abbreviation or a specific instance (for example, inst.1 or inst.2).
filepath	oval-def:EntityObjectStringType (1..1)	The filepath element specifies the absolute path for a file or directory in the specified package.
oval-def:filter	n/a (0..unbounded)	

< packagecheck_state >

The package_state element defines the different verification information associated with SVR4 packages installed on the system. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1210: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
pkginst	oval-def:EntityStateStringType (0..1)	The pkginst entity is a string that represents a package designation by its instance. An instance can be the package abbreviation or a specific instance (for example, inst.1 or inst.2).
filepath	oval-def:EntityStateStringType (0..1)	The filepath element specifies the absolute path for a file or directory in the specified package.
checksum_differs	oval-def:EntityStateBoolType (0..1)	Has the file's checksum changed? A value of true indicates that the file's checksum has changed. A value of false indicates that the file's checksum has not changed.
size_differs	oval-def:EntityStateBoolType (0..1)	Has the file's size changed? A value of true indicates that the file's size has changed. A value of false indicates that the file's size has not changed.
mtime_differs	oval-def:EntityStateBoolType (0..1)	Has the file's modified time changed? A value of true indicates that the file's modified time has changed. A value of false indicates that the file's modified time has not changed.
uread	sol-def:EntityStatePermissionCompareType (0..1)	Has the actual user read permission changed from the expected user read permission?
uwrite	sol-def:EntityStatePermissionCompareType (0..1)	Has the actual user write permission changed from the expected user write permission?
uexec	sol-def:EntityStatePermissionCompareType (0..1)	Has the actual user exec permission changed from the expected user exec permission?
gread	sol-def:EntityStatePermissionCompareType (0..1)	Has the actual group read permission changed from the expected group read permission?
gwrite	sol-def:EntityStatePermissionCompareType (0..1)	Has the actual group write permission changed from the expected group write permission?
gexec	sol-def:EntityStatePermissionCompareType (0..1)	Has the actual group exec permission changed from the expected group exec permission?
oread	sol-def:EntityStatePermissionCompareType (0..1)	Has the actual others read permission changed from the expected others read permission?
owrite	sol-def:EntityStatePermissionCompareType (0..1)	Has the actual others read permission changed from the expected others read permission?
oexec	sol-def:EntityStatePermissionCompareType (0..1)	Has the actual others read permission changed from the expected others read permission?

== PackageCheckBehaviors ==

The PackageCheckBehaviors complex type defines a set of behaviors that for controlling how installed SVR4 packages are checked. These behaviors align with the options of the pkgchk command (specifically ‘-a’, ‘-c’, and ‘-n’).

Attributes

Table 1211: Attributes

Attribute	Type	Desc.
fileattributes_only	xsd:boolean (optional *default*=’false’)	‘fileattributes_only’ when true this behavior means only check the file attributes and do not check file contents. When false, both file attributes and contents will be checked. This aligns with the pkgchk option ‘-a’.
filecontents_only	xsd:boolean (optional *default*=’false’)	‘filecontents_only’ when true this behavior means only check the file contents and do not check file attributes. When false, both file attributes and contents will be checked. This aligns with the pkgchk option ‘-c’.
no_volatileeditable	xsd:boolean (optional *default*=’false’)	‘no_volatileeditable’ when true this behavior means do not check volatile or editable files’ contents. When false, volatile and editable files’ contents will be checked. This aligns with the pkgchk option ‘-n’.

< packagefreezelist_test >

The packagefreezelist_test provides support for checking the metadata of IPS packages that have been frozen at a particular version. The test extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a packagefreezelist_object and the optional state elements reference packagefreezelist_states that specify the metadata to check about a set of packages.

Extends: oval-def:TestType

Child Elements

Table 1212: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< packagefreezelist_object >

The packagefreezelist_object element is used by a packagefreezelist_test to identify the set of IPS packages that have been frozen at a particular version on a system. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

< packagefreezelist_state >

The packagefreezelist_state element defines the different system state information that can be used to evaluate the specified IPS packages on a Solaris system that have been frozen at a particular version.

Extends: oval-def:StateType

Child Elements

Table 1213: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
fmri	oval-def:EntityStateStringType (0..1)	The Fault Management Resource Identifier (FMRI) of the package which uniquely identifies the package on the system.

< packagepublisher_test >

The packagepublisher_test provides support for checking the metadata of package publishers on a Solaris system. The test extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a packagepublisher_object and the optional state elements reference packagepublisher_states that specify the metadata to check about a set of package publishers on a Solaris system.

Extends: oval-def:TestType

Child Elements

Table 1214: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< packagepublisher_object >

The packagepublisher_object element is used by a packagepublisher_test to identify the set of package publishers to check on a Solaris system. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 1215: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityObjectStringType (1..1)	The name of the IPS package publisher.
type	sol-def:EntityObjectPublisherTypeType (1..1)	The type of the IPS package publisher.
origin_uri	oval-def:EntityObjectStringType (0..1)	The origin URI of the IPS package publisher.
oval-def:filter	n/a (0..unbounded)	

< packagepublisher_state >

The packagepublisher_state element defines the different system information that can be used to evaluate the specified package publishers.

Extends: oval-def:StateType

Child Elements

Table 1216: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-def:EntityStateStringType (0..1)	The name of the IPS package publisher.
type	sol-def:EntityStatePublisherTypeType (0..1)	The type of the IPS package publisher.
origin_uri	oval-def:EntityStateStringType (0..1)	The origin URI of the IPS package publisher.
alias	oval-def:EntityStateStringType (0..1)	The alias of the IPS package publisher.
ssl_key	oval-def:EntityStateStringType (0..1)	The Secure Socket Layer (SSL) key registered by a client for publishers using client-side SSL authentication.
ssl_cert	oval-def:EntityStateStringType (0..1)	The Secure Socket Layer (SSL) certificate registered by a client for publishers using client-side SSL authentication.
client_uuid	sol-def:EntityStateClientUUIDType (0..1)	The universally unique identifier (UUID) that identifies the image to its IPS package publisher.
catalog_updated	oval-def:EntityStateIntType (0..1)	The last time that the IPS package publisher's catalog was updated in seconds since the Unix epoch. The Unix epoch is the time 00:00:00 UTC on January 1, 1970.
enabled	oval-def:EntityStateBoolType (0..1)	Specifies whether or not the IPS package publisher is enabled.
order	oval-def:EntityStateIntType (0..1)	Specifies where in the search order the IPS package publisher is listed. The first publisher in the search order will have a value of '1'.
properties	oval-def:EntityStateRecordType (0..1)	The properties associated with the IPS package publisher.

< patch54_test >

The patch test is used to check information associated with different patches for SVR4 packages installed on the system. Image Packaging System (IPS) packages do not support patches and are not supported by this test. The information being tested is based off the /usr/bin/showrev -p command. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an inetd_object and the optional state element specifies the information to check.

Extends: oval-def:TestType

Child Elements

Table 1217: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< patch_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.4
- Reason: Replaced by the patch54_test. The new test includes additional functionality that allows the object element to match both the original patch and any superseding patches. As a result of this new functionality, the patch_object was also expanded to include behaviors and version entities. See the patch54_test.
- Comment: This test has been deprecated and will be removed in version 6.0 of the language.

The patch test is used to check information associated with different patches installed on the system. The information being tested is based off the /usr/bin/showrev -p command. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an inetd_object and the optional state element specifies the information to check.

Extends: oval-def:TestType

Child Elements

Table 1218: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< patch54_object >

The patch54_object element is used by a patch test to define the specific patch to be evaluated. Patches are identified by unique alphanumeric strings, with the patch base code first, a hyphen, and a number that represents the patch revision number. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A patch object consists of a base entity that identifies the patch to be used, and a version entity that represent the patch revision number.

Extends: oval-def:ObjectType

Child Elements

Table 1219: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	sol-def:PatchBehaviors (0..1)	
base	oval-def:EntityObjectIntType (1..1)	The base entity represents a patch base code found before the hyphen.
version	oval-def:EntityObjectIntType (1..1)	The version entity represents a patch version number found after the hyphen.
oval-def:filter	n/a (0..unbounded)	

< patch_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.4
- Reason: Replaced by the patch54_object. Due to the additional functionality that allows the object element to match both the original patch and any superseding patches, a new object was created that includes behaviors and version entities. See the patch54_object.
- Comment: This object has been deprecated and will be removed in version 6.0 of the language.

The patch_object element is used by a patch test to define the specific patch to be evaluated. Patches are identified by unique alphanumeric strings, with the patch base code first, a hyphen, and a number that represents the patch revision number. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A patch object consists of a single base entity that identifies the patch to be used.

Extends: oval-def:ObjectType

Child Elements

Table 1220: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
base	oval-def:EntityObjectIntType (1..1)	The base entity reresents a patch base code found before the hyphen.

< patch_state >

The patch_state element defines the different information associated with a specific patch for an SVR4 package installed on the system. Patches are identified by unique alphanumeric strings, with the patch base code first, a hyphen, and a number that represents the patch revision number. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1221: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
base	oval-def:EntityStateIntType (0..1)	The base entity reresents a patch base code found before the hyphen.
version	oval-def:EntityStateIntType (0..1)	The version entity represents a patch version number found after the hyphen.

== PatchBehaviors ==

The PatchBehaviors complex type defines a number of behaviors that allow a more detailed definition of the patch_object being specified. 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.

Attributes

Table 1222: Attributes

Attribute	Type	Desc.
supersedence	Restriction of xsd:boolean (optional *default*='false')	'supersedence' specifies that the object should also match any superseding patches to the one being specified. In Solaris, a patch can be superseded in two ways. The first way is implicitly when a new revision of a patch is released (e.g. patch 12345-02 supersedes patch 12345-01). The second way is explicitly where a new patch contains the complete functionality of another patch. If set to 'true', the resulting object set would be the original patch specified plus any superseding patches. The default value is 'false' meaning the object should only match the specified patch.

< smf_test >

The smf_test is used to check service management facility controlled services including traditional unix rc level start/kill scrips and inetd daemon services. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a smf_object and the optional state element specifies the information to check.

Extends: oval-def:TestType

Child Elements

Table 1223: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< smf_object >

The smf_object element is used by a smf_test to define the specific service instance to be evaluated. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A smf_object consists of a fmri entity that represents the Fault Management Resource Identifier (FMRI) which uniquely identifies a service.

Extends: oval-def:ObjectType

Child Elements

Table 1224: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
fmri	oval-def:EntityObjectStringType (1..1)	The FMRI (Fault Managed Resource Identifier) entity is used to identify system objects for which advanced fault and resource management capabilities are provided. Services managed by SMF are assigned FMRI URIs prefixed with the scheme name “svc”. FMRIs used by SMF can be expressed in three ways: first as an absolute path including a location path such as “localhost” (eg svc://localhost/system/system-log:default), second as a path relative to the local machine (eg svc:/system/system-log:default), and third as simply the service identifier with the string prefixes implied (eg system/system-log:default). For OVAL, the absolute path version (first choice) should be used.
oval-def:filter	n/a (0..unbounded)	

< smf_state >

The smf_state element defines the different information associated with a specific smf controlled service. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1225: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
fmri	oval-def:EntityStateStringType (0..1)	The FMRI (Fault Managed Resource Identifier) entity describes a possible identifier associated with a service. Services managed by SMF are assigned FMRI URIs prefixed with the scheme name “svc”. FMRIs used by SMF can be expressed in three ways: first as an absolute path including a location path such as “localhost” (eg svc://localhost/system/system-log:default), second as a path relative to the local machine (eg svc:/system/system-log:default), and third as simply the service identifier with the string prefixes implied (eg system/system-log:default). For OVAL, the absolute path version (first choice) should be used.
service_name	oval-def:EntityStateStringType (0..1)	The service_name entity is usually an abbreviated form of the FMRI. In the example svc://localhost/system/system-log:default, the name would be system-log.
service_state	sol-def:EntityStateSmfServiceStateType (0..1)	The service_state entity describes a possible state that the service may be in. Each service instance is always in a well-defined state based on its dependencies, the results of the execution of its methods, and its potential receipt of events from the contracts filesystem. The service_state values are UNINITIALIZED, OFFLINE, ONLINE, DEGRADED, MAINTENANCE, DISABLED, and LEGACY-RUN.
protocol	oval-def:EntityStateStringType (0..1)	The protocol entity describes a possible protocol supported by the service.
server_executable	oval-def:EntityStateStringType (0..1)	The entity server_executable is a string representing the listening daemon on the server side. An example being ‘svcprop ftp’ which might show ‘inetd/start/exec astring /usr/sbin/in.ftpd-a’
server_arguments	oval-def:EntityStateStringType (0..1)	The server_arguments entity describes possible parameters that are passed to the service.
exec_as_user	oval-def:EntityStateStringType (0..1)	The exec_as_user entity is a string pulled from svcprop in the following format: inetd_start/user astring root

---

< smfproperty_test >

The smfproperty_test is used to check the value of properties associated with SMF services. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an smfproperty_object and the optional state elements reference a smfproperty_state and specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 1226: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< smfproperty_object >

The smfproperty_object element is used by a SMF property test to define the SMF property items to be evaluated based on the specified states. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 1227: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
service	oval-def:EntityObjectStringType (1..1)	Specifies the SMF service on the system. This is the service category and name separated by a forward slash ("/").
instance	oval-def:EntityObjectStringType (1..1)	The instance of an SMF service which represents a specific configuration of a service.
property	oval-def:EntityObjectStringType (1..1)	The name of the property associated with an SMF service. This is the property category and name separated by a forward slash ("/").
oval-def:filter	n/a (0..unbounded)	

< smfproperty_state >

The smfproperty_state specifies the values of properties associated with SMF services.

Extends: oval-def:StateType

Child Elements

Table 1228: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
service	oval-def:EntityStateStringType (0..1)	Specifies the SMF service on the system. This is the service category and name separated by a forward slash ("/").
instance	oval-def:EntityStateStringType (0..1)	Specifies the instance of an SMF service which represents a specific configuration of a service.
property	oval-def:EntityStateStringType (0..1)	Specifies the name of the property associated with an SMF service. This is the property category and name separated by a forward slash ("/").
fmri	oval-def:EntityStateStringType (0..1)	The Fault Management Resource Identifier (FMRI) of the SMF service which uniquely identifies the service on the system.
value	oval-def:EntityStateAnySimpleType (0..1)	Specifies the value of the property associated with an SMF service.

< variant_test >

The variant_test is used to check the variants associated with the current Image Packaging System image. Variants are properties that control whether or not mutually exclusive components from a package are installed on a system. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an variant_object and the optional state elements reference a variant_state and specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 1229: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< variant_object >

The variant_object element is used by a variant test to define the image variant items to be evaluated based on the specified states. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Extends: oval-def:ObjectType

Child Elements

Table 1230: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
path	oval-def:EntityObjectStringType (1..1)	The path to the Solaris IPS image.
name	oval-def:EntityObjectStringType (1..1)	The name of the variant property associated with an IPS image.
oval-def:filter	n/a (0..unbounded)	

< variant_state >

The variant_state specifies the various variant properties associated with the specified IPS image.

Extends: oval-def:StateType

Child Elements

Table 1231: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
path	oval-def:EntityStateStringType (0..1)	Specifies the path to the Solaris IPS image.
name	oval-def:EntityStateStringType (0..1)	Specifies the name of the variant property associated with an IPS image.
value	oval-def:EntityStateAnySimpleType (0..1)	Specifies the value of the variant property associated with an IPS image.

< virtualizationinfo_test >

The virtualizationinfo_test provides support for checking the metadata associated with the current virtualization environment this instance of Solaris is running on. The test extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a virtualizationinfo_object and the optional state elements reference virtualizationinfo_states that specify the metadata to check the current virtualization environment.

Extends: oval-def:TestType

Child Elements

Table 1232: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< virtualizationinfo_object >

The virtualizationinfo_object element is used by a virtualizationinfo_test to identify the current virtualization environment this instance of Solaris is running on. Given that this object only retrieves the current virtualization environment for the system, there are no child entities to specify in the object.

Extends: oval-def:ObjectType

< virtualizationinfo_state >

The virtualizationinfo_state element defines the different information that can be used to evaluate the current virtualization environment this instance of Solaris is running on.

Extends: oval-def:StateType

Child Elements

Table 1233: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
current	oval-def:EntityStateStringType (0..1)	The name of the current environment.
supported	sol-def:EntityStateV12NEnvType (0..1)	The list of virtualization environments that this node supports as children.
parent	sol-def:EntityStateV12NEnvType (0..1)	The parent environment of the current environment.
ldom-role	sol-def:EntityStateLDOMRoleType (0..1)	The logical domain roles associated with the current environment.
properties	oval-def:EntityStateRecordType (0..1)	The properties associated with the current environment.

== EntityObjectPublisherTypeType ==

The EntityObjectPublisherTypeType complex type restricts a string value to three values: archive, mirror, or origin that specifies how the publisher distributes their packages. The empty string is also allowed to support empty elements associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityObjectStringType

Table 1234: Enumeration Values

Value	Description
archive	The value of 'archive' specifies that the publisher distributes packages by providing a file that contains one or more packages.
mirror	The value of 'mirror' specifies that the publisher distributes packages by providing a package repository that contains only package content.
origin	The value of 'origin' specifies that the publisher distributes packages by providing a package repository that contains both package metadata and package content.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateClientUUIDType ==

The EntityStateClientUUIDType restricts a string value to a representation of a client UUID, used to identify an image to its IPS package publisher. The empty string is also allowed to support empty element associated with variable references. 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.

Restricts: oval-def:EntityStateStringType

Pattern: ([a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{12})?

== EntityStatePermissionCompareType ==

The EntityStatePermissionCompareType complex type restricts a string value to more, less, or same which specifies if an actual permission is different than the expected permission (more or less restrictive) or if the permission is the same. The empty string is also allowed to support empty elements associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 1235: Enumeration Values

Value	Description
more	The actual permission is more restrictive than the expected permission.
less	The actual permission is less restrictive than the expected permission.
same	The actual permission is the same as the expected permission.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStatePublisherTypeType ==

The EntityStatePublisherTypeType complex type restricts a string value to three values: archive, mirror, or origin that specifies how the publisher distributes their packages. The empty string is also allowed to support empty elements associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 1236: Enumeration Values

Value	Description
archive	The value of ‘archive’ specifies that the publisher distributes packages by providing a file that contains one or more packages.
mirror	The value of ‘mirror’ specifies that the publisher distributes packages by providing a package repository that contains only package content.
origin	The value of ‘origin’ specifies that the publisher distributes packages by providing a package repository that contains both package metadata and package content.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateSmfServiceStateType ==

The EntityStateSmfServiceStateType complex type defines the different values that are valid for the service_state entity of a smf_state. The empty string is also allowed as a valid value to support an empty element that is found when a variable reference is used within the type entity.

Restricts: oval-def:EntityStateStringType

Table 1237: Enumeration Values

Value	Description
DEGRADED	The instance is enabled and running or available to run. The instance, however, is functioning at a limited capacity in comparison to normal operation.
DISABLED	The instance is disabled.
MAINTENANCE	The instance is enabled, but not able to run. Administrative action is required to restore the instance to offline and subsequent states.
LEGACY-RUN	This state represents a legacy instance that is not managed by the service management facility. Instances in this state have been started at some point, but might or might not be running.
OFFLINE	The instance is enabled, but not yet running or available to run.
ONLINE	The instance is enabled and running or is available to run.
UNINITIALIZED	This is the initial state for all service instances.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateV12NEnvType ==

The EntityStateV12NEnvType complex type restricts a string value to a specific set of values that describe the virtalization environment. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 1238: Enumeration Values

Value	Description
unknown	The virtualization environment is unknown. This could mean it is a bare metal virtualization environment.
kvm	The virtualization environment is a Kernel-based Virtual Machine (KVM).
logical-domain	The virtualization environment is a logical domain.
non-global-zone	The virtualization environment is a non-global zone.
kernel-zone	The virtualization environment is a kernel zone.
vmware	The virtualization environment is VMware.
virtualbox	The virtualization environment is Oracle VirtualBox.
xen	The virtualization environment is Xen.
	The empty string value is permitted here to allow for empty elements associated with variable references.

== EntityStateLDOMRoleType ==

The EntityStateLDOMRoleType complex type restricts a string value to a specific set of roles for the current virtualization environment. The empty string is also allowed to support empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 1239: Enumeration Values

Value	Description
control-role	The current virtualization environment is a control domain.
io-role	The current virtualization environment is an I/O domain.
root-role	The current virtualization environment is a root I/O domain.
service-role	The current virtualization environment is a service domain.
	The empty string value is permitted here to allow for empty elements associated with variable references.

Open Vulnerability and Assessment Language: Solaris System Characteristics

- Schema: Solaris System Characteristics
- Version: 5.11.1:1.1
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the Solaris specific system characteristic items found in Open Vulnerability and Assessment Language (OVAL). Each item is an extension of the standard test element defined in the Core Definition Schema. Through extension, each test inherits a set of elements and attributes that are shared amongst all OVAL tests. Each test is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core Definition Schema is not outlined here.

The OVAL Schema is maintained by the 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.cisecurity.org.

Item Listing

- *< facet_item >*
- *< image_item >*
- *< isainfo_item >*
- *< ndd_item >*

- *< package_item >*
- *< package511_item >*
- *< packageavoidlist_item >*
- *< packagecheck_item >*
- *< packagefreezelist_item >*
- *< packagepublisher_item >*
- *< patch_item >*
- *< smf_item >*
- *< smfproperty_item >*
- *< variant_item >*
- *< virtualizationinfo_item >*

< facet_item >

This item stores the facet properties and values of an IPS system image.

Extends: oval-sc:ItemType

Child Elements

Table 1240: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
path	oval-sc:EntityItemStringType (0..1)	Specifies the path to the Solaris IPS image.
name	oval-sc:EntityItemStringType (0..1)	Specifies the name of the facet property associated with an IPS image.
value	oval-sc:EntityItemBoolType (0..1)	Specifies the value of the facet property associated with an IPS image.

< image_item >

This item stores system state information associated with an IPS image on a Solaris system.

Extends: oval-sc:ItemType

Child Elements

Table 1241: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
path	oval-sc:EntityItemStringType (0..1)	The path to the Solaris IPS image.
name	oval-sc:EntityItemStringType (0..1)	The name of the property associated with the Solaris IPS image.
value	oval-sc:EntityItemAnySimpleType (0..unbounded)	The value of a property that is associated with a Solaris IPS image.

< isainfo_item >

Information about the instruction set architectures. This information can be retrieved by the isainfo command.

The isainfo_item was originally developed by Robert L. Hollis at ThreatGuard, Inc. Many thanks for their support of the OVAL project.

Extends: oval-sc:ItemType

Child Elements

Table 1242: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
bits	oval-sc:EntityItemIntType (0..1)	This is the number of bits in the address space of the native instruction set (isainfo -b).
kernel_isa	oval-sc:EntityItemStringType (0..1)	This is the name of the instruction set used by kernel components (isainfo -k).
application_isa	oval-sc:EntityItemStringType (0..1)	This is the name of the instruction set used by portable applications (isainfo -n).

< ndd_item >

This item represents data collected by the ndd command.

Extends: oval-sc:ItemType

Child Elements

Table 1243: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
device	oval-sc:EntityItemStringType (0..1)	The name of the device for which the parameter was collected.
instance	oval-sc:EntityItemIntType (0..1)	The instance of the device to examine. Certain devices may have multiple instances on a system. If multiple instances exist, this entity should be populated with its respective instance value. If only a single instance exists, this entity should not be collected.
parameter	oval-sc:EntityItemStringType (0..1)	The name of a parameter for example, ip_forwarding
value	oval-sc:EntityItemAnySimpleType (0..1)	The observed value of the named parameter.

< package_item >

The package_item holds information about installed SVR4 packages. Output of /usr/bin/pkginfo. See pkginfo(1).

Extends: oval-sc:ItemType

Child Elements

Table 1244: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
pkginst	oval-sc:EntityItemStringType (0..1)	
name	oval-sc:EntityItemStringType (0..1)	
category	oval-sc:EntityItemStringType (0..1)	
version	oval-sc:EntityItemStringType (0..1)	
vendor	oval-sc:EntityItemStringType (0..1)	
description	oval-sc:EntityItemStringType (0..1)	

< package511_item >

This item stores system state information associated with IPS packages installed on a Solaris system.

Extends: oval-sc:ItemType

Child Elements

Table 1245: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
publisher	oval-sc:EntityItemStringType (0..1)	The person, group of persons, or organization that is the source of the package. The publisher should be expressed without leading "pkg:" or "//" components.
name	oval-sc:EntityItemStringType (0..1)	The full hierarchical name of the package which is separated by forward slash characters. The full name should be expressed without leading "pkg:/" or "/" components.
version	oval-sc:EntityItemVersionType (0..1)	The version of the package which consists of the component version, build version, and branch version.
timestamp	oval-sc:EntityItemStringType (0..1)	The timestamp when the package was published in the ISO-8601 basic format (YYYYMMDDTHHMMSSZ).
fmri	oval-sc:EntityItemStringType (0..1)	The Fault Management Resource Identifier (FMRI) of the package which uniquely identifies the package on the system.
summary	oval-sc:EntityItemStringType (0..1)	A summary of what the package provides.
description	oval-sc:EntityItemStringType (0..1)	A description of what the package provides.
category	oval-sc:EntityItemStringType (0..1)	The category of the package.
updates_available	oval-sc:EntityItemBoolType (0..1)	A boolean value indicating whether or not updates are available for this package.

< packageavoidlist_item >

This item stores the FMRI associated with associated with IPS packages that have been flagged as to be avoided from installation on a Solaris system.

Extends: oval-sc:ItemType

Child Elements

Table 1246: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
fmri	oval-sc:EntityItemStringType (0..1)	The Fault Management Resource Identifier (FMRI) of the package which uniquely identifies the package on the system.

< packagecheck_item >

The packagecheck_item holds verification information about an individual file that is part of an installed SVR4 package. Each packagecheck_item contains a package designation, filepath, whether the checksum differs, whether the size differs, whether the modfication time differs, and how the actual permissions differ from the expected permissions. For more information, see pkgchk(1M). It extends the standard ItemType as defined in the oval-system-characteristics schema and one should refer to the ItemType description for more information.

Extends: oval-sc:ItemType

Child Elements

Table 1247: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
pkginst	oval-sc:EntityItemStringType (0..1)	The pkginst entity is a string that represents a package designation by its instance. An instance can be the package abbreviation or a specific instance (for example, inst.1 or inst.2).
filepath	oval-sc:EntityItemStringType (0..1)	The filepath element specifies the absolute path for a file or directory in the specified package..
checksum_differs	oval-sc:EntityItemBoolType (0..1)	Has the file's checksum changed? A value of true indicates that the file's checksum has changed. A value of false indicates that the file's checksum has not changed.
size_differs	oval-sc:EntityItemBoolType (0..1)	Has the file's size changed? A value of true indicates that the file's size has changed. A value of false indicates that the file's size has not changed.
mtime_differs	oval-sc:EntityItemBoolType (0..1)	Has the file's modified time changed? A value of true indicates that the file's modified time has changed. A value of false indicates that the file's modified time has not changed.
uread	sol-sc:EntityItemPermissionCompareType (0..1)	Has the actual user read permission changed from the expected user read permission?
uwrite	sol-sc:EntityItemPermissionCompareType (0..1)	Has the actual user write permission changed from the expected user write permission?
uexec	sol-sc:EntityItemPermissionCompareType (0..1)	Has the actual user exec permission changed from the expected user exec permission?
gread	sol-sc:EntityItemPermissionCompareType (0..1)	Has the actual group read permission changed from the expected group read permission?
gwrite	sol-sc:EntityItemPermissionCompareType (0..1)	Has the actual group write permission changed from the expected group write permission?
gexec	sol-sc:EntityItemPermissionCompareType (0..1)	Has the actual group exec permission changed from the expected group exec permission?
oread	sol-sc:EntityItemPermissionCompareType (0..1)	Has the actual others read permission changed from the expected others read permission?
owrite	sol-sc:EntityItemPermissionCompareType (0..1)	Has the actual others read permission changed from the expected others read permission?
oexec	sol-sc:EntityItemPermissionCompareType (0..1)	Has the actual others read permission changed from the expected others read permission?

< packagefreezelist_item >

This item stores the FMRI associated with associated with IPS packages that have been frozen at a particular version.

Extends: oval-sc:ItemType

Child Elements

Table 1248: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
fmri	oval-sc:EntityItemStringType (0..1)	The Fault Management Resource Identifier (FMRI) of the package which uniquely identifies the package on the system.

< packagepublisher_item >

This item stores system state information associated with IPS package publishers on a Solaris system.

Extends: oval-sc:ItemType

Child Elements

Table 1249: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
name	oval-sc:EntityItemStringType (0..1)	The name of the IPS package publisher.
type	sol-sc:EntityItemPublisherTypeType (0..1)	The type of the IPS package publisher.
origin_uri	oval-sc:EntityItemStringType (0..1)	The origin URI of the IPS package publisher.
alias	oval-sc:EntityItemStringType (0..1)	The alias of the IPS package publisher.
ssl_key	oval-sc:EntityItemStringType (0..1)	The Secure Socket Layer (SSL) key registered by a client for publishers using client-side SSL authentication.
ssl_cert	oval-sc:EntityItemStringType (0..1)	The Secure Socket Layer (SSL) certificate registered by a client for publishers using client-side SSL authentication.
client_uuid	sol-sc:EntityItemClientUUIDType (0..1)	The universally unique identifier (UUID) that identifies the image to its publisher.
catalog_updated	oval-sc:EntityItemIntType (0..1)	The last time that the IPS package publisher's catalog was updated in seconds since the Unix epoch. The Unix epoch is the time 00:00:00 UTC on January 1, 1970.
enabled	oval-sc:EntityItemBoolType (0..1)	Specifies whether or not the publisher is enabled.
order	oval-sc:EntityItemIntType (0..1)	Specifies where in the search order the IPS package publisher is listed. The first publisher in the search order will have a value of '1'.
properties	oval-sc:EntityItemRecordType (0..1)	The properties associated with an IPS package publisher.

< patch_item >

Patches for SVR4 packages are identified by unique alphanumeric strings, with the patch base code first, a hyphen, and a number that represents the patch revision number. The information can be obtained using /usr/bin/showrev -p. Please see showrev(1M).

Extends: oval-sc:ItemType

Child Elements

Table 1250: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
base	oval-sc:EntityItemIntType (0..1)	The base entity reresents a patch base code found before the hyphen.
version	oval-sc:EntityItemIntType (0..1)	The version entity represents a patch version number found after the hyphen.

< smf_item >

The smf_item is used to hold information related to service management facility controlled services

Extends: oval-sc:ItemType

Child Elements

Table 1251: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
fmri	oval-sc:EntityItemStringType (0..1)	The FMRI (Fault Managed Resource Identifier) entity holds the identifier associated with a service. Services managed by SMF are assigned FMRI URIs prefixed with the scheme name “svc”. FMRIs used by SMF can be expressed in three ways: first as an absolute path including a location path such as “localhost” (eg svc://localhost/system/system-log:default), second as a path relative to the local machine (eg svc:/system/system-log:default), and third as simply the service identifier with the string prefixes implied (eg system/system-log:default). For OVAL, the absolute path version (first choice) should be used.
service_name	oval-sc:EntityItemStringType (0..1)	The service_name entity is usually an abbreviated form of the FMRI. In the example svc://localhost/system/system-log:default, the name would be system-log.
service_state	sol-sc:EntityItemSmfServiceStateType (0..1)	The service_state entity describes the state that the service is in. Each service instance is always in a well-defined state based on its dependencies, the results of the execution of its methods, and its potential receipt of events from the contracts filesystem. The service_state values are UNINITIALIZED, OFFLINE, ONLINE, DEGRADED, MAINTENANCE, DISABLED, and LEGACY-RUN.
protocol	oval-sc:EntityItemStringType (0..unbounded)	The protocol entity describes the protocol supported by the service.
server_executable	oval-sc:EntityItemStringType (0..1)	The entity server_executable is a string representing the listening daemon on the server side. An example being ‘svcprop ftp’ which might show ‘inetd/start/exec astring /usr/sbin/in.ftpd-a’
server_arguments	oval-sc:EntityItemStringType (0..1)	The server_arguments entity describes the parameters that are passed to the service.
exec_as_user	oval-sc:EntityItemStringType (0..1)	The exec_as_user entity is a string pulled from svcprop in the following format: inetd_start/user astring root

< smfproperty_item >

This item stores the properties and values of an SMF service.

Extends: oval-sc:ItemType

Child Elements

Table 1252: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
service	oval-sc:EntityItemStringType (0..1)	Specifies the SMF service on the system. This is the service category and name separated by a forward slash ("/").
instance	oval-sc:EntityItemStringType (0..1)	Specifies the instance of an SMF service which represents a specific configuration of a service.
property	oval-sc:EntityItemStringType (0..1)	The name of the property associated with an SMF service. This is the property category and name separated by a forward slash ("/").
fmri	oval-sc:EntityItemStringType (0..1)	The Fault Management Resource Identifier (FMRI) of the SMF service which uniquely identifies the service on the system.
value	oval-sc:EntityItemAnySimpleType (0..1)	Specifies the value of the property associated with an SMF service.

< variant_item >

This item stores the variant properties and values of the specified IPS system image.

Extends: oval-sc:ItemType

Child Elements

Table 1253: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
path	oval-sc:EntityItemStringType (0..1)	Specifies the path to the Solaris IPS image.
name	oval-sc:EntityItemStringType (0..1)	Specifies the name of the variant property associated with an IPS image.
value	oval-sc:EntityItemAnySimpleType (0..unbounded)	Specifies the value of the variant property associated with an IPS image.

< virtualizationinfo_item >

This item stores the information associated with the current virtualization environment this instance of Solaris is running on and is capable of supporting.

Extends: oval-sc:ItemType

Child Elements

Table 1254: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
current	oval-sc:EntityItemStringType (0..1)	The name of the current environment. This information could be collected using the libv12n library or by executing the 'virtinfo -c current list -H -o name' command.
supported	sol-sc:EntityItemV12NEnvType (0..unbounded)	The list of virtualization environments that this node supports as children. This information could be collected using the libv12n library or by executing the 'virtinfo -c supported list -H -o name' command.
parent	sol-sc:EntityItemV12NEnvType (0..1)	The parent environment of the current environment. This information could be collected using libv12n library or by executing the 'virtinfo -c parent list -H -o name' command.
ldom-role	sol-sc:EntityItemLDOMRoleType (0..unbounded)	The logical domain roles associated with the current environment. This information could be collected using libv12n library.
properties	oval-sc:EntityItemRecordType (0..1)	The properties associated with the current environment. This information could be collected using libv12n library.

== EntityItemClientUUIDType ==

The EntityItemClientUUIDType restricts a string value to a representation of a client UUID, used to identify an image to its IPS package publisher. The empty string is also allowed to support empty element associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Pattern: ([a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{12})?

== EntityItemPermissionCompareType ==

The EntityItemPermissionCompareType complex type restricts a string value to more, less, or same which specifies if an actual permission is different than the expected permission (more or less restrictive) or if the permission is the same. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 1255: Enumeration Values

Value	Description
more	The actual permission is more restrictive than the expected permission.
less	The actual permission is less restrictive than the expected permission.
same	The actual permission is the same as the expected permission.
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemPublisherTypeType ==

The EntityItemPublisherTypeType complex type restricts a string value to three values: archive, mirror, or origin that specifies how the publisher distributes their packages. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 1256: Enumeration Values

Value	Description
archive	The value of ‘archive’ specifies that the publisher distributes packages by providing a file that contains one or more packages.
mirror	The value of ‘mirror’ specifies that the publisher distributes packages by providing a package repository that contains only package content.
origin	The value of ‘origin’ specifies that the publisher distributes packages by providing a package repository that contains both package metadata and package content.
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemSmfServiceStateType ==

The EntityItemSmfServiceStateType defines the different values that are valid for the service_state entity of a smf_item. The empty string is also allowed as a valid value to support empty emlements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 1257: Enumeration Values

Value	Description
DEGRADED	The instance is enabled and running or available to run. The instance, however, is functioning at a limited capacity in comparison to normal operation.
DISABLED	The instance is disabled.
MAINTENANCE	The instance is enabled, but not able to run. Administrative action is required to restore the instance to offline and subsequent states.
LEGACY-RUN	This state represents a legacy instance that is not managed by the service management facility. Instances in this state have been started at some point, but might or might not be running.
OFFLINE	The instance is enabled, but not yet running or available to run.
ONLINE	The instance is enabled and running or is available to run.
UNINITIALIZED	This is the initial state for all service instances.
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemV12NEnvType ==

The EntityItemV12NEnvypeType complex type restricts a string value to a specific set of values that describe the virtalization environment. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 1258: Enumeration Values

Value	Description
unknown	The virtualization environment is unknown. This could mean it is a bare metal virtualization environment.
kvm	The virtualization environment is a Kernel-based Virtual Machine (KVM).
logical-domain	The virtualization environment is a logical domain.
non-global-zone	The virtualization environment is a non-global zone.
kernel-zone	The virtualization environment is a kernel zone.
vmware	The virtualization environment is VMware.
virtualbox	The virtualization environment is Oracle VirtualBox.
xen	The virtualization environment is Xen.
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemLDOMRoleType ==

The EntityItemLDOMRoleType complex type restricts a string value to a specific set of roles for the current virtualization environment. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 1259: Enumeration Values

Value	Description
control-role	The current virtualization environment is a control domain.
io-role	The current virtualization environment is an I/O domain.
root-role	The current virtualization environment is a root I/O domain.
service-role	The current virtualization environment is a service domain.
	The empty string value is permitted here to allow for detailed error reporting.

Open Vulnerability and Assessment Language: VMware ESX server Definition

- Schema: VMware ESX server Definition
- Version: 5.11.1:1.1
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the VMware ESX server specific tests found in Open Vulnerability and Assessment Language (OVAL). Each test is an extension of the standard test element defined in the Core Definition Schema. Through extension, each test inherits a set of elements and attributes that are shared amongst all OVAL tests. Each test is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core Definition Schema is not outlined here.

This schema was originally developed by Yuzheng Zhou and Todd Dolinsky at Hewlett-Packard. The OVAL Schema is maintained by the 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.cisecurity.org.

Test Listing

- *< patch56_test >*
- *< patch_test > (Deprecated)* (Deprecated)
- *< version_test >*

- *< visdkmanagedobject_test >*

< patch56_test >

The patch56_test reveals the installation status of a specific patch or patches in VMware ESX Server. This information can be retrieved by the "esxupdate query" command. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a patch56_object and the optional state element referencing a patch56_state specifies the metadata to check.

Note that different from previous versions, ESX Server 3.0.3 and ESX Server 3.5 use the following patch naming convention: {ProductName}{VersionNumber}-{BundleID}-{Classification}{SupportLevel}. Please refer to http://www.vmware.com/pdf/vi3_35/esx_3/r35/vi3_35_25_esxupdate.pdf for more detailed information.

Extends: oval-def:TestType

Child Elements

Table 1260: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< patch56_object >

The patch56_object element is used by a patch56_test to define those objects to be evaluated against a specified state. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A patch56_object consists of a single patch_name entity that identifies the patch to be checked.

Extends: oval-def:ObjectType

Child Elements

Table 1261: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	esx-def:Patch56Behaviors (0..1)	
patch_name	oval-def:EntityObjectStringType (1..1)	The patch name entity indetifies a specific patch or set of patches to be checked on the system. For example: ESX-200603 or ESX350-200904401-BG. The value of this entity should correspond to the values returned under the "name" column of the "esxupdate query" command.
oval-def:filter	n/a (0..unbounded)	

< patch56_state >

The patch56_state element defines the different information that can be used to evaluate the specified VMware ESX Serer patch. Please refer to the individual elements in the schema for more details about what each represents.

Extends: oval-def:StateType

Child Elements

Table 1262: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
patch_name	oval-def:EntityStateStringType (0..1)	The patch_name entity indetifies the name of a patch to test for. For example: ESX-200603 or ESX350-200904401-BG. The value of this entity should correspond to the values returned under the “name” column of the “esxupdate query” command.
knowledge_base_id	oval-def:EntityStateIntType (0..1)	The knowledge_base_id entity specifies a given knowledge base article identifier number. This entity is valid for ESX versions 3.0.2 and earlier. It is comprised of the numerical string at the end of the patch name. For example, the patch ESX-200603 would have a knowledge base identifier of 200603.
bundle_id	oval-def:EntityStateIntType (0..1)	The bundle_id entity specifies a unique ID for the patch. This entity is valid for ESX version 3.0.3 and version 3.5 and is comprised of the year and month the bundle was released and a 3-digit unique ID. It is in the format YYYYMM###. For example, the first patch released in January 2008 might have a BundleID of 200801001.
classification	esx-def:EntityStateClassificationType (0..1)	The classification entity specifies the type of patch. It can be one of: B - bug, U - update, S - security, or R - roll-up. This entity is valid for ESX version 3.0.3 and later.
support_level	esx-def:EntityStateSupportLevelType (0..1)	The support_level entity specifies a support level to test for. If can be one of: G - GA patch, H - hot patch, D - debugging patch, or C - custom patch. This entity is valid for ESX version 3.0.3 and later.
status	oval-def:EntityStateBoolType (0..1)	The status entity specifies an installation status of a patch to test for. A value of ‘true’ is used to signify that a given patch is intalled.

== Patch56Behaviors ==

The Patch56Behaviors complex type defines a number of behaviors that allow a more detailed definition of the patch56_object being specified. 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.

Attributes

Table 1263: Attributes

Attribute	Type	Desc.
supersedence	Restriction of xsd:boolean (optional *default*='false')	'supersedence' specifies that the object should also match any superseding patches to the one being specified. In other words, if set to True the resulting object set would be the original patch specified plus any superseding patches. The default value is 'false' meaning the object should only match the specified patch.

< patch_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.6
- Reason: Replaced by the patch56_test. The deprecated patch_test has a bug where the patch name entity is defined as a string in the object yet is defined as an int in the state. Additional state entities have also been added to the new patch56_test.
- Comment: This test has been deprecated and will be removed in version 6.0 of the language.

The patch test reveals the installation status of a specific patch in the VMware ESX server. This information can be retrieved by the "esxupdate query | grep ESX-xxxxxxx" command. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a patch_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 1264: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< patch_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.6
- Reason: Replaced by the patch56_object. The deprecated patch_test has a bug where the patch name entity is defined as a string in the object yet is defined as an int in the state. Additional state entities have also been added to the new patch56_test.
- Comment: This object has been deprecated and will be removed in version 6.0 of the language.

The patch_object element is used by a patch test to define those objects to be evaluated based on a specified state. Each object extends the standard ObjectType as defined in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

A patch_object consists of a single patch_number entity that identifies the patch to be checked.

Extends: oval-def:ObjectType

Child Elements

Table 1265: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	esx-def:PatchBehaviors (0..1)	
patch_number	oval-def:EntityObjectStringType (1..1)	The patch_number entity identifies the patch to be checked. Many of the security bulletins for VMWARE ESX Server contain non-numerical characters in the patch number, therefore this entity has a datatype of string.

< patch_state > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.6
- Reason: Replaced by the patch56_state. The deprecated patch_test has a bug where the patch name entity is defined as a string in the object yet is defined as an int in the state. Additional state entities have also been added to the new patch56_test.
- Comment: This object has been deprecated and will be removed in version 6.0 of the language.

The patch_state element defines the information about a specific patch. The patch_number element identifies this patch, and the status element reveals the installation status of this patch in the VMware ESX server. For instance, after the “esxupdate query | grep ESX-2559638” command is run, the result is either a string similar to “ESX-2559638 15:27:17 04/05/07 Update info rpm for ESX 3.0.1.” or empty.

Extends: oval-def:StateType

Child Elements

Table 1266: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
patch_number	oval-def:EntityStateStringType (0..1)	This is the patch number of a specific patch which will be checked in current VMware ESX server. Many of the security bulletins for VMWARE ESX Server contain non-numerical characters in the patch nubmer, therefore this entity has a datatype of string.
status	oval-def:EntityStateBoolType (0..1)	This is the installation status of a specific patch in current VMware ESX server.

== PatchBehaviors == (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.6
- Reason: Replaced by Patch56Behaviors. The deprecated patch_test has a bug where the patch name entity is defined as a string in the object yet is defined as an int in the state. Additional state entities have also been added to the new patch56_test.
- Comment: These behaviors have been deprecated and will be removed in version 6.0 of the language.

The PatchBehaviors complex type defines a number of behaviors that allow a more detailed definition of the patch_object being specified. 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.

Attributes

Table 1267: Attributes

Attribute	Type	Desc.
supersedence	Restriction of xsd:boolean (optional *default*='false')	'supersedence' specifies that the object should also match any superseding patches to the one being specified. In other words, if set to True the resulting object set would be the original patch specified plus any superseding patches. The default value is 'false' meaning the object should only match the specified patch.

< version_test >

The version test reveals information about the release and build version of the VMware ESX server. This information can be retrieved by the "vmware -v" command or by checking the /proc/vmware/version file. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a version_object and the optional state element specifies the metadata to check.

Extends: oval-def:TestType

Child Elements

Table 1268: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< version_object >

The version_object element is used by a version test to define those objects to be evaluated based on a specified state. There is actually only one object relating to version and this is the ESX server as a whole. Therefore, there are no child entities defined. Any OVAL Test written to check version will reference the same version_object which is basically an empty object element.

Extends: oval-def:ObjectType

< version_state >

The version_state element defines the information about the release and build version. The release and build elements specify the release and build information of the VMware ESX server respectively. For instance, if the output of "vmware -v" command is "VMware ESX Server 3.0.1 build-39823", then release is equal to "3.0.1" and build is equal to "39823".

Extends: oval-def:StateType

Child Elements

Table 1269: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
release	oval-def:EntityStateVersionType (0..1)	This is the release version of current VMware ESX server.
build	oval-def:EntityStateIntType (0..1)	This is the build version of current VMware ESX server.

< visdkmanagedobject_test >

The visdkmanagedobject_test is used to check information about Managed Objects in the VMware Infrastructure. This test extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references a visdkmanagedobject _object and the optional state element specifies the metadata to check.

This test has been introduced to enable standardized automated assessments of configuration settings in cloud computing components. All aspects of the VMware cloud can be considered in this test due to the VMware Infrastructure. Whether it is a Virutal Machine, a Host System, or even a Data Center, properties are defined in ways that can be enumerated in a common methodology. The VI SDK Programming Guide located at http://www.vmware.com/support/developer/vc-sdk/visdk400pubs/sdk40programmingguide.pdf serves as a great resource. Chapter 3 discusses the Managed Entities enumerated in the behaviors.

There are several Managed Entities in the VMware Infrastructure which have been enumerated in ViSdkManagedEntityBehaviors to enable interpreters to execute efficient interrogations. This test is designed for an interpreter to access Managed Entity properties (settings) via the VI SDK webservice. An example use case is to interrogate all virtual machines to ensure that a particular security setting is enabled. Some properties serve to configure the Virtual Machine, while others can be used to identify. For example, sets and filters can be used to create a set of all Virtual Machines where bridged networking is employed, and then perform an OVAL state evaluation against each of those Virtual Machines. This concept applies to all properties across all Managed Entities. Use the ViSdkManagedEntityBehaviors to avoid enumerating all Managed Objects when only one type should be considered.

Extends: oval-def:TestType

Child Elements

Table 1270: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< visdkmanagedobject_object >

The visdkmanagedobject_object element is used by the visdkmanagedobject_test to define those objects to be evaluated based on a specified state.

Extends: oval-def:ObjectType

Child Elements

Table 1271: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
behaviors	esx-def:ViSdkManagedEntityBehaviors (0..1)	
property	oval-def:EntityObjectStringType (1..1)	The property entity holds a string that represents the object path path and name of a particular setting for the Managed Entity. In the VMware Infrastructure SDK, property names are case-sensitive and thus case must be correct relative to the properties in the SDK. For example, a Virtual Machine might have ethernet0.connectionType of 'bridged'.
oval-def:filter	n/a (0..unbounded)	

< visdkmanagedobject_state >

The visdkmanagedobject_state elements enumerates the different properties a Managed Entity might have. Managed Entities have the same object structure. However, fields within that object structure will be blank (null) if they do not apply to that Managed Entity.

Extends: oval-def:StateType

Child Elements

Table 1272: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
property	oval-def:EntityStateStringType (0..1)	The property entity holds a string that represents the object path and name of a particular setting for the Managed Entity. In the VMware Infrastructure SDK, property names are case-sensitive and thus case must be correct relative to the properties in the SDK. For example, a Virtual Machine might have ethernet0.connectionType of 'bridged'.
value	oval-def:EntityStateAnySimpleType (0..1)	The value entity holds a string that represents a value that's associated with the specified setting for the Managed Entity. Some properties will return an array of values. In such cases consider each value individually and then make final evaluation based on the entity_check attribute.

== ViSdkManagedEntityBehaviors ==

The ViSdkManagedEntityBehaviors complex type defines a number of behaviors that allow a more detailed definition of the visdkmanagedobject_object being specified. Note that using these behaviors is *highly* encouraged because enumerating all Managed Objects in an inventory hierarchy could cause performance problems. Interpreters should enumerate only the entities specified by the behavior prior to set/filter logic and evaluation.

Attributes

Table 1273: Attributes

Attribute	Type	Desc.
managed_entity_type	Restriction of xsd:string (optional *default*='VirtualMachine') ('ClusterComputerResource', 'ComputeResource', 'Datacenter', 'Datastore', 'DistributedVirtualPortgroup', 'DistributedVirtualSwitch', 'Folder', 'HostSystem', 'Network', 'ResourcePool', 'VirtualApp', 'VirtualMachine')	The 'managed_entity_type' defines the type of managed object from which the property and value should be collected.

== EntityStateClassificationType ==

The EntityStateClassificationType complex type restricts a string value to a specific set of values that describe the classification of a given ESX Server patch. The empty string is also allowed to support an empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 1274: Enumeration Values

Value	Description
B	Bug patches fix minor flaws that affect product functionality or behavior. Bug patches are optional. Before they are applied, one should determine whether they are necessary for your environment.
R	Roll-up patches contain any number of bundles for ESX Server 3.0.3 or ESX Server 3.5 hosts. They can contain bug patches, update patches, and security patches. They do not contain upgrade bundles for minor releases or update bundles for maintenance releases.
S	Security patches fix one or more potential security vulnerabilities in the product. They should be implemented immediately to prevent the vulnerabilities from being exploited.
U	Update patches can contain new driver updates and small non-intrusive enhancements. Before they are applied, one should determine whether they are necessary for your environment.
	The empty string is also allowed to support an empty element associated with variable references.

== EntityStateSupportLevelType ==

The EntityStateSupportLevelType complex type restricts a string value to a specific set of values that describe the support level of a given ESX Server patch. The empty string is also allowed to support an empty element associated with variable references. Note that when using pattern matches and variables care must be taken to ensure that the regular expression and variable values align with the enumerated values.

Restricts: oval-def:EntityStateStringType

Table 1275: Enumeration Values

Value	Description
C	Custom patches are special fixes provided to a customer. They are usually specific to customer's environment, and are most likely not required by customers not reporting the issue. Custom patches have been tested in the customer's environment.
D	Debugging patches are released to all customers and are used by VMware to troubleshoot complex product issues. They can contain debug messages and code, and drivers. Debugging patches usually require VMware assistance to install.
G	GA patches are released to all customers and have been thoroughly tested. They contain fixes for ESX Server 3 software issues.
H	Hot patches are released to specific customers for solving critical problems specific to their environment. They contain fixes for security issues or problems that can potentially cause data loss or severe service disruptions. Hot patches should be implemented immediately.
	The empty string is also allowed to support an empty element associated with variable references.

Open Vulnerability and Assessment Language: VMware ESX server System Characteristics

- Schema: VMware ESX server System Characteristics
- Version: 5.11.1:1.1
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the VMware ESX server specific system characteristic items found in Open Vulnerability and Assessment Language (OVAL). Each item is an extension of the standard test element defined in the Core Definition Schema. Through extension, each test inherits a set of elements and attributes that are shared amongst all OVAL tests. Each test is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core Definition Schema is not outlined here.

This schema was originally developed by Yuzheng Zhou and Todd Dolinsky at Hewlett-Packard. The OVAL Schema is maintained by the 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.cisecurity.org.

Item Listing

- *< patch_item >*
- *< version_item >*
- *< visdkmanagedobject_item >*

< patch_item >

Installation information about a specific patch in the VMware ESX server. This information can be retrieved by the "esxupdate query | grep ESX-xxxxxxx" command.

Extends: oval-sc:ItemType

Child Elements

Table 1276: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
patch_number (Deprecated)	oval-sc:EntityItemStringType (0..1)	This is the patch number which identifies the patch being checked in current VMware ESX server. Many of the security bulletins for VMWARE ESX Server contain non-numerical characters in the patch number, therefore this entity has a datatype of string.
patch_name	oval-sc:EntityItemStringType (0..1)	The patch_name entity indetifies the name of the patch. For example: ESX-200603 or ESX350-200904401-BG. The value of this entity should correspond to the values returned under the "name" column of the "esxupdate query" command.
knowledge_base_id	oval-sc:EntityItemIntType (0..1)	The knowledge_base_id entity specifies the knowledge base article identifier number associated with a given patch from ESX versions 3.0.2 and earlier. It is comprised of the numerical string at the end of the patch name. For example, the patch ESX-200603 would have a knowledge base identifier of 200603. For patches from ESX version 3.0.3 and later, the patch name uses a different format and does not include the knowledge base id. This entity should be marked with a status of 'does not exist' in those cases.
bundle_id	oval-sc:EntityItemIntType (0..1)	The bundle_id entity specifies the unique ID for the patch. Note that for version 3.0.3 and version 3.5 this is comprised of the year and month the bundle was released and a 3-digit unique ID. It is in the format YYYYMM###. For example, the first patch released in January 2008 might have a BundleID of 200801001. For patches from ESX version 3.0.2 and earlier, this entity should be marked with a status of 'does not exist' since patch name has a different format and doesn't include a bundle id.
classification	esx-sc:EntityItemClassificationType (0..1)	The classification entity specifies the type of patch. It can be one of: B - bug, U - update, S - security, or R - rollup. For patches from ESX version 3.0.2 and earlier, this entity should be marked with a status of 'does not exist' since patch name has a different format and doesn't include a classification.
support_level	esx-sc:EntityItemSupportLevelType (0..1)	The support_level entity specifies the support level of the patch. If can be one of: G - GA patch, H - hot patch, D - debugging patch, or C - custom patch. For patches from ESX version 3.0.2 and earlier, this entity should be marked with a status of 'does not exist' since patch name has a different format and doesn't include a support level.
status	oval-sc:EntityItemBoolType (0..1)	This is the installtaion status of the specific patch.

< version_item >

Information about the release and build version of VMware ESX server. This information can be retrieved by the "vmware -v" command or by checking the /proc/vmware/version file.

Extends: oval-sc:ItemType

Child Elements

Table 1277: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
release	oval-sc:EntityItemVersionType (0..1)	This is the release of current VMware ESX server.
build	oval-sc:EntityItemIntType (0..1)	This is the build version of current VMware ESX server.

< visdkmanagedobject_item >

The visdkmanagedobject_item is used to represent information about Managed Objects in the VMware Infrastructure.

Extends: oval-sc:ItemType

Child Elements

Table 1278: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
property	oval-sc:EntityItemStringType (0..1)	The property entity holds a string that represents the object path and name of a particular setting for the Managed Entity. In the VMware Infrastructure SDK, property names are case-sensitive and thus case must be correct relative to the properties in the SDK. For example, a Virtual Machine might have ethernet0.connectionType of 'bridged'.
value	oval-sc:EntityItemAnySimpleType (0..unbounded)	The value entity holds a string that represents a value that's associated with the specified setting for the Managed Entity. Some properties will return an array of values. In such cases consider each value individually and then make final evaluation based on the entity_check attribute.

== EntityItemClassificationType ==

The EntityItemClassificationType complex type restricts a string value to a specific set of values that describe the classification of a given ESX Server patch. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 1279: Enumeration Values

Value	Description
B	Bug patches fix minor flaws that affect product functionality or behavior. Bug patches are optional. Before they are applied, one should determine whether they are necessary for your environment.
R	Roll-up patches contain any number of bundles for ESX Server 3.0.3 or ESX Server 3.5 hosts. They can contain bug patches, update patches, and security patches. They do not contain upgrade bundles for minor releases or update bundles for maintenance releases.
S	Security patches fix one or more potential security vulnerabilities in the product. They should be implemented immediately to prevent the vulnerabilities from being exploited.
U	Update patches can contain new driver updates and small non-intrusive enhancements. Before they are applied, one should determine whether they are necessary for your environment.
	The empty string value is permitted here to allow for detailed error reporting.

== EntityItemSupportLevelType ==

The EntityItemSupportLevelType complex type restricts a string value to a specific set of values that describe the support level of a given ESX Server patch. The empty string is also allowed to support empty elements associated with error conditions.

Restricts: oval-sc:EntityItemStringType

Table 1280: Enumeration Values

Value	Description
C	Custom patches are special fixes provided to a customer. They are usually specific to customer's environment, and are most likely not required by customers not reporting the issue. Custom patches have been tested in the customer's environment.
D	Debugging patches are released to all customers and are used by VMware to troubleshoot complex product issues. They can contain debug messages and code, and drivers. Debugging patches usually require VMware assistance to install.
G	GA patches are released to all customers and have been thoroughly tested. They contain fixes for ESX Server 3 software issues.
H	Hot patches are released to specific customers for solving critical problems specific to their environment. They contain fixes for security issues or problems that can potentially cause data loss or severe service disruptions. Hot patches should be implemented immediately.
	The empty string value is permitted here to allow for detailed error reporting.

Open Vulnerability and Assessment Language: Apache Definition

- Schema: Apache Definition
- Version: 5.11.1:1.1
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the Apache specific tests found in Open Vulnerability and Assessment Language (OVAL). Each test is an extension of the standard test element defined in the Core Definition Schema. Through extension, each test inherits a set of elements and attributes that are shared amongst all OVAL tests. Each test is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core Definition Schema is not outlined here.

The OVAL Schema is maintained by the 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.cisecurity.org.

Test Listing

- *< httpd_test > (Deprecated)* (Deprecated)

< httpd_test > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.8
- Reason: The httpd_test does not specify how to detect instances of httpd and cannot be reasonably specified to allow for products to detect all instances of httpd across platforms, packaging systems, and typical user compiled and configured installations. Without a proper definition of how to identify instances of httpd products will not reliably produce consistent assessment results because they will naturally utilize different approaches to locating instances of httpd which will lead to differences in the set of collected instances of https.
- Comment: This test has been deprecated and may be removed in a future version of the language.

The httpd test is used to check the version of an installed httpd binary. It extends the standard TestType as defined in the oval-definitions-schema and one should refer to the TestType description for more information. The required object element references an httpd_test and the optional state element specifies the data to check.

Extends: oval-def:TestType

Child Elements

Table 1281: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
object	oval-def:ObjectRefType (1..1)	
state	oval-def:StateRefType (0..unbounded)	

< httpd_object > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.8
- Reason: The httpd_object does not specify how to detect instances of httpd and cannot be reasonably specified to allow for products to detect all instances of httpd across platforms, packaging systems, and typical user compiled and configured installations. Without a proper definition of how to identify instances of httpd products will not reliably produce consistent assessment results because they will naturally utilize different approaches to locating instances of httpd which will lead to differences in the set of collected instances of https.
- Comment: This object has been deprecated and may be removed in a future version of the language.

The httpd_object element is used by a httpd test to define the different httpd binary installed on a system. There is actually only one object relating to this and it is the collection of all httpd binaries. Therefore, there are no child entities defined. Any OVAL Test written to check version will reference the same httpd_object which is basically an empty object element. A tool that implements the httpd_test and collects the httpd_object must know how to find all the httpd binaries on the system and verify that they are in fact httpd binaries.

Extends: oval-def:ObjectType

< httpd_state > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.8
- Reason: The httpd_state does not specify how to detect instances of httpd and cannot be reasonably specified to allow for products to detect all instances of httpd across platforms, packaging systems, and typical user compiled and configured installations. Without a proper definition of how to identify instances of httpd products will not reliably produce consistent assessment results because they will naturally utilize different approaches to locating instances of httpd which will lead to differences in the set of collected instances of https.
- Comment: This state has been deprecated and may be removed in a future version of the language.

The httpd_state element defines information associated with a specific httpd binary.

Extends: oval-def:StateType

Child Elements

Table 1282: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
path	oval-def:EntityStateStringType (0..1)	The path element specifies the directory component of the absolute path to a httpd binary on the system.
binary_name	oval-def:EntityStateStringType (0..1)	The binary_name element specifies the name of the file. If the xsi:nil attribute is set to true, then the object being specified is the higher level path. In this case, the binary_name element should not be collected or used in analysis. Setting xsi:nil equal to true is different than using a .* pattern match, says to collect every file under a given path.
version	oval-def:EntityStateVersionType (0..1)	The version entity is used to check the version of the httpd binary. The datatype for the version entity is 'version' which means the value should be a delimited set of numbers. It is obtained by running 'httpd -v'.

Open Vulnerability and Assessment Language: Apache System Characteristics

- Schema: Apache System Characteristics
- Version: 5.11.1:1.1
- Release Date: 11/30/2016 09:00:00 AM

The following is a description of the elements, types, and attributes that compose the Apache specific system characteristic items found in Open Vulnerability and Assessment Language (OVAL). Each item is an extension of the standard item element defined in the Core System Characteristic Schema. Through extension, each item inherits a set of elements and attributes that are shared amongst all OVAL Items. Each item is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core System Characteristic Schema is not outlined here.

The OVAL Schema is maintained by the 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.cisecurity.org.

Item Listing

- *< httpd_item > (Deprecated)*

< httpd_item > (Deprecated)

Deprecation Info

- Deprecated As Of Version 5.8
- Reason: The httpd_item does not specify how to detect instances of httpd and cannot be reasonably specified to allow for products to detect all instances of httpd across platforms, packaging systems, and typical user compiled and configured installations. Without a proper definition of how to identify instances of httpd products will not reliably produce consistent assessment results because they will naturally utilize different approaches to locating instances of httpd which will lead to differences in the set of collected instances of https.
- Comment: This item has been deprecated and may be removed in a future version of the language.

The httpd item holds information about a installed Apache HTTPD binary. It extends the standard ItemType as defined in the oval-system-characteristics schema and one should refer to the ItemType description for more information.

Extends: oval-sc:ItemType

Child Elements

Table 1283: Elements

Child Elements	Type (MinOccurs..MaxOccurs)	Desc.
path	oval-sc:EntityItemStringType (0..1)	The path element specifies the directory component of the absolute path to a httpd binary found on the system.
binary_name	oval-sc:EntityItemStringType (0..1)	The name of the httpd binary.
version	oval-sc:EntityItemVersionType (0..1)	The version entity holds the version of the specified httpd binary.

5.3 OVAL Design Principles

Requirements language in this document are defined in RFC 2119. Design principles are categorized as generally applicable or applicable to the versions as indicated. An update mechanism is built into the language development process to account for the fact that new design principles may be desired in the future.

5.3.1 General OVAL Design Princples

High-Level

- Capabilities SHOULD NOT require changes to scanned systems in order to implement (e.g. be read-only)
- Changes SHOULD NOT impose security issues
- Changes SHOULD NOT result in inconsistency
- Changes SHOULD NOT require obsoleted technologies or methodologies
- Changes SHOULD NOT require the use of undocumented APIs
- Changes SHOULD NOT duplicate existing capabilities unless there is a compelling reason to do so (e.g. major simplification), in which case they should be designed to allow for the deprecation and ultimate replacement of the constructs that are duplicated
- OVAL capabilities should fit into the OVAL use cases
- Capabilities SHOULD NOT dictate implementation, but should document at least one practical implementation method

Construct-Specific

- An OVAL Item MUST model the posture attribute data being collected off an endpoint
- An OVAL Item MUST NOT combine multiple system-level structures
- OVAL Items SHOULD only include the OVAL Entities that the community requires
- An OVAL Object SHOULD include the minimum set of OVAL Entities needed to uniquely identify an OVAL Item collected from an endpoint

Mechanics (naming, versioning, etc.)

- Changes SHOULD NOT break backwards compatibility within their major version (i.e. 5.x, 6.x, etc.)
- OVAL constructs MUST conform to naming conventions (http://ovalproject.github.io/getting-started/best-practices/#4-naming-conventions)
- OVAL constructs MUST follow the versioning policy

5.3.2 OVAL Use Cases

- **Security Advisory Distribution:** allows application and operating system vendors to release advisories in a machine-readable format, moving authoring of technical details of a vulnerability from second-hand (e.g. scanner product developers) to first-hand (product developers)
- **Vulnerability Assessment:** increases transparency into vulnerability management process, quality of checks, and ease of feature comparison between tools

- **Patch Management:** allows patch management vendors to quickly consume data from multiple sources
- **Configuration Management:** eliminates the need for IT professionals to translate paper configuration documents into something that can be applied and enforced
- **System Inventory:** shifts burden of inventory definition from best guesses by system inventory tool vendors to authoritative knowledge sources including application and operating system authors
- **Malware Artifact Hunting:** provides a standardized format for encoding malware artifacts
- **Network Access Control (NAC):** standardizes policy expression for policy checking and enforcement when an endpoint requests access to a network and on an ongoing basis for continued policy conformance
- **Auditing and Centralized Audit Validation:** captures machine configuration information that allows organizations to monitor, track, and reconstruct the transition of a system's configuration from one state to another
- **Security Information Management Systems:** simplifies the interoperability of SIMS with a standardized data exchange format

5.4 OVAL Specifications

These specifications detail the current version of OVAL, as well as the UNIX and Windows extensions. They can all be downloaded as .docx files.

- `OVAL Language Specification`
- `OVAL UNIX Extension Specification`
- `OVAL Windows Extension Specification`

5.5 Community Organization

Note that requirements language used within this section is defined by RFC2119.

The OVAL Community includes:

- *Community Members*: Responsible for maintaining OVAL and these governance processes by creating issues, reviewing issues, creating change proposals, collaborating on change proposals, reviewing change proposals and generally contributing to this consensus-driven process.
- *OVAL Leadership Board*: Steers the OVAL mission and use cases, assists (when needed) with consensus calls, is instrumental in updating design principles, and is responsible for selecting the Official Release(s).
- *Area Supervisors*: An Area Supervisor is responsible for the day-to-day management of its appointed area.
- *OVAL Sponsor*: Mainly handles logistics for managing OVAL resources, managing area supervisor appointments, operating the OVAL Repository, and so on.

5.5.1 How the Community Works

The gist of OVAL Community operations is basically that any Community Member can make a proposal about anything OVAL-related at any time, and that proposal follows through our proposal process. Community Members, Area Supervisors, the Sponsor, and the Leadership Board may all play a role in the process as a proposal moves from start to finish.

Community Members

Community Members are responsible for maintaining OVAL and these governance processes by creating issues, reviewing issues, creating change proposals, collaborating on change proposals, reviewing change proposals and generally contributing to this consensus-driven process.

Join Us

Joining the OVAL Community is free and all you need to do is join one of our *OVAL Mailing Lists* and make a contribution, whether that be asking questions, answering questions, making language proposals or contributions of other kinds. Proposals and issues can best be raised through Github, which requires an account. If you don't have one, set one up here and start contributing!

OVAL Leadership Board

In general, members of the OVAL Leadership Board are:

- Responsible for steering the OVAL Mission, Use Cases and providing guidance to other key strategic artifacts including the OVAL Design Principles
- Expected to monitor project activities and weigh in on issues facing the Community, especially in areas in which they have special expertise
- Expected to offer guidance to the Community when called on by the Supervisors and/or Community Members, as outlined by the language development process (see "Language Governance Responsibilities" below)
- Expected to engage in online and in-person events subject to their availability
- Expected to advocate on behalf of the project to the general public when appropriate
- Expected to participate in quarterly calls and vote when votes are called

Leadership Board Members

Current Members

- Organizational
  - Arctic Wolf - David Solin, David Ries
  - Center for Internet Security - William Munyan, Adam Montville
  - Cisco Systems, Inc. - Omar Santos
  - HCL Group - Rosario Gangemi
  - McAfee - Kent Landfield
  - Modulo - Alberto Bastos
  - National Institute of Standards and Technology - Stephen Banghart
  - Qualys, Inc. - Hariom Singh
  - Red Hat, Inc. - Matej Tyc, Watson Sato
  - SecPod Technologies - Chandrashekhar B
  - NAVWARSYSCOM, U.S. Navy - Jack Vander Pol

  - Unified Compliance - Stephen Pillero
  - VMware - Dennis Moreau
- Individual
  - Blake Frantz
  - Dale Rich
  - Nils Puhlmann

Past Members

- Emeritus
  - Anton Chuvakin
  - Javier Fernandez-Sanguino
  - Jay Beale
  - Mark Cox
  - Matt Hansbury
  - Robert Hollis
  - Tim "TK" Keanini
- Former
  - Amol Sarwate
  - Anthony Busciglio
  - Carl Banzhof
  - Chris Wood
  - Eric Walker
  - Gary Miliefsky
  - Jamie Cromer
  - Jay Graver
  - Kurt Seifried
  - Luigi Pichetti
  - Martin Preisler
  - Melissa Albanese
  - Michael Tan
  - Morey Haber
  - Nick Connor
  - Noah Salzman
  - Panos Kampanakis
  - Pat Fetty
  - Randy Taylor

  - Stephen Quinn

Responsibilities

Language Governance Responsibilities

The OVAL Leadership Board plays a role in key aspects of the overall language development process.

Language Development

In overall language development, the OVAL Leadership Board has an important, direct function in providing guiding support to a consensus call, when needed. Such support may be needed when consensus is too difficult for an Area Supervisor to judge. The IETF has published an excellent informational RFC on the subject: RFC 7282.

Update Design Principles

As proposals to the language are received in the language development process, we may find a need to update the OVAL Design Principles. A specific subprocess has been defined to handle this case, where the OVAL Leadership Board plays an important role.

- **Review Proposed Update:** All parties review the proposed update to the design principles
- **Suggest Change:** All parties are able to suggest changes to the proposed update
- **Create Consensus Call:** The leadership board MUST formally call for consensus on the design principle update proposal
- **Address Issue:** When issues are raised during a formal consensus call, the Leadership Board MUST acknowledge and take appropriate action for the raised issue
- **Update Design Principles:** The Leadership board MUST authorize updates to the design principles (another party, i.e. the sponsor, may actually update the design principles artifact)

Official OVAL Release

The OVAL Leadership Board is responsible for selecting a Stable releass at least once per year to be the Official release.

Processes

Membership

New Members

New members of the OVAL Leadership Board are nominated by one or more existing members. Appointment to the board is confirmed by a vote.[1] The Sponsor will facilitate such votes in a timely basis.

[1] OVAL Board members participating during the time MITRE was the OVAL Sponsor have been carried forward as initial members of the Leadership Board.

Recognition of Former Members

Former OVAL Leadership Board members will be considered for recognition by the Sponsor under the following guidelines:

- Emeritus Member: a person who made significant contributions to this community
- Former Contributing Member: a person who made clear contributions to this community

If a person did not make a measurable contribution to this community, then the person is not identified as a former member.

Changing Roles in an Organization

If a current OVAL Leadership Board member switches roles within an organization and serving on the Board no longer makes sense, they must notify the Sponsor. Upon notification, the member will be given an opportunity to nominate a new member to represent the organization. This prospective member will be considered in accordance with the New Members process.

Leaving an Organization

If a current OVAL Leadership Board member is going to leave an organization, they must notify the Sponsor. Upon notification, the current member will be given two options:

- They can continue to serve on the Board under their new organization.
- They can relinquish their membership and will be considered for recognition as a former member as described under Recognition of Former Members.

In either case, the organization that is losing representation on the OVAL Leadership Board will be given an opportunity to nominate a new member that will be considered in accordance with the New Members Process.

Revocation of Membership

If the Sponsor has evidence that an OVAL Leadership Board member is not fulfilling their responsibilities, they may be removed. The following process defines the steps that the Sponsor must follow in order to revoke the membership of a current member.

- The Sponsor must provide the member with a warning of revocation at least two (2) months before revocation is scheduled to occur explaining the reasons for revocation.
- The Sponsor may delay the date of revocation.
- Prior to revocation, the member will be given an opportunity to get in good standing according to the agreed upon responsibilities. If membership no longer makes sense, it will be terminated.
- If the member fails to get in good standing, their membership will be revoked and they will not be recognized as a former member.

Voting

What Is Voted On?

The OVAL Leadership Board will be required to vote on the following matters.

- Approval of an official OVAL release
- Approval of new OVAL Leadership Board members

Lastly, a vote may be requested for any other issue deemed necessary by the OVAL Leadership Board or the Sponsor. Each request will be considered on a case-by-case by the Sponsor to see if it is within the Board's responsibilities as described herein. If a request falls within one of these areas, the request will be processed and a vote will be announced. To request a vote, a member can either publicly send a message to the Board mailing list or privately send a message to the Sponsor.

Who May Vote?

All active members of the OVAL Leadership Board are eligible to cast a vote. However, only one vote per organization will be accepted. Emeritus members are not eligible to cast a vote, but, they can provide their input on matters before a decision is made.

Announcing a Vote

All matters, which require a vote, will be announced on the Board mailing list and the OVAL developer mailing list along with the timeline. The timeline will provide a deadline for community and Board discussion as well as dates for when the voting period begins and ends.

Casting a Vote

All voting ballots will be distributed through email over the Board mailing list and will typically require that an organization select one or more options as well as provide justification. Please note that all votes and justifications will be posted to the OVAL Community repository to provide the community with transparency into the voting process and for record-keeping purposes.

Handling Multiple Votes from an Organization

In the event that multiple, conflicting votes are cast by the same organization, only the first vote received will count. If all members of the affected organization reaching consensus on changing a vote, they may request their vote be changed by emailing the Board mailing list before voting has closed. The Sponsor will consider the reasons for changing the vote and determine which of the votes should be considered valid. Please note that any changes to a vote will be considered on a case-by-case basis and should only be approved given extenuating circumstances.

Total Possible Votes

Because only one vote may be accepted per organization, the total number of possible votes equals the number of distinct organizations having organizational members plus the number of individual members.

Quorum

In order to reach a quorum, votes must be cast by a simple majority of the Total Possible Votes. If a quorum is not reached, a vote will be deemed invalid.

Reaching a Decision

A decision is reached if there is a quorum and the results of the vote indicate that a simple majority of the votes are for or against a particular issue. If there is a tie, the Sponsor will re-open the discussion and schedule another vote on the issue.

Publishing Vote Results

Once the OVAL Leadership Board reaches a decision, the results of the vote will be announced over the Board mailing list and the OVAL developer mailing list, and posted to the OVAL Community repository.

Area Supervisors

An Area Supervisor is responsible for the day-to-day management of its appointed area.

Area Supervisors

The following list bounds the different areas of responsibility that may be managed by a given Area Supervisor.

- Foundation Schemas (i.e. Core, Common, Independent, Results, Variables and Directives Schemas) - Bill Munyan, CIS
- Android Schemas - Pooja Shetty, SecPod
- Apple Schemas (iOS, Macintosh) - David Solin, Joval
- Cisco Schemas (ASA, CatOS, IOS, IOS XE, PixOS) - Omar Santos, Cisco
- FreeBSD Schemas - David Solin, Joval
- HP-UX Schemas - John Ulmer, US Navy SPAWAR
- IBM AIX Schemas - John Ulmer, US Navy SPAWAR
- Linux Schemas - Watson Sato, Red Hat
- UNIX Schemas - Watson Sato, Red Hat
- Sun Solaris Schemas - Jarrett Lu, Oracle
- Juniper JunOS Schemas - David Solin, Joval
- NETCONF Schemas - David Solin, Joval
- Microsoft Schemas (i.e. Microsoft Windows, SharePoint) - Jack Vander Pol, US Navy SPAWAR
- Vmware Schemas (i.e. ESX) - TBD

Responsibilities

Individuals or organizations appointed to be an Area Supervisor are appointed to one or more Areas of Responsibility. In general, each Supervisor MUST:

- Be engaged with the language development process
- Consult with other Supervisors where appropriate
- Ensure coverage for their area(s) in the event of an expected absence

Language Governance Responsibilities

Area Supervisors have can important role to play throughout the language development process, as they are involved in nearly all of its facets.

Language Development

- Consensus Call: See "Consensus Building" below
- Release: The Supervisor is responsible for releasing updates to the OVAL Language within its purview, which may require coordination with other Area Supervisors

Update Design Principles

- Review Proposed Update: All parties review the proposed update to the design principles
- Suggest Change: All parties are able to suggest changes to the proposed update

Consensus Building

- Update Design Principles: See "Update Design Principles" above
- Create Consensus Call: The Area Supervisor MUST formally call for consensus for the proposal under review in their area
- Address Issue: When issues are raised during a formal consensus call, the Area Supervisor MUST acknowledge and take appropriate action for the raised issue

Appointment

Area Supervisors are appointed by the Sponsor.

OVAL Sponsor

The Sponsor is essentially the conservator of the OVAL Language, providing infrastructure, logistical, and community support to further the language.

Sponsor

The Center for Internet Security is the presently designated sponsor.

Responsibilities

General Responsibilities

In general, the Sponsor will:

- **Function as the OVAL account owner:** own all project resources such as GitHub account containing repos, mailing list accounts, static site, etc.

- **Manage Area Supervisor appointments:** ensure that each Area of Responsibility has at least one Area Supervisor.

Language Governance Responsibilities

The Sponsor has a less critical role in governing the OVAL Language than other roles. Note that the individual(s)/organization with the Sponsor role may play additional roles in the language development process.

Language Development

- **Logistical Support:** The Sponsor provides logistical support during the release process.
- **Release Announcement:** The Sponsor is responsible for announcing OVAL Language releases.

Appointment

In the event of the Sponsor relinquishing their role, they will nominate a new Sponsor to fill the role. If they do not nominate a replacement, the issue will be brought to the Leadership Board to select a new Sponsor.

5.6 Proposal Process

The Proposal Process is the way we collaborate to improve OVAL, including:

- Fixing a defect in the OVAL schemas or documentation
- Adding new features, tests, or platforms
- Updating these Guidelines (e.g. the Developer Guides, Content Repository Listings, etc.)
- Updating this Proposal Process itself!

In fact, the only things that are NOT updated via the Proposal Process are:

- The *OVAL Leadership Board*, *Area Supervisors* and *OVAL Sponsor* (follow links to learn more)
- The OVAL Mission, Use Cases and *OVAL Design Principles* (these are maintained by the *OVAL Leadership Board*)

5.6.1 Process Overview

Here is a quick rundown of the process. Please click through the links to read more about each phase.

1. *Create an Issue*: anyone may create an Issue describing a problem or suggesting an enhancement
2. *Initial Proposal*: anyone may submit a Proposal for resolving the Issue so the community can review and provide feedback
3. *Alternate Proposals*: community members may submit Alternate Proposals for resolving the Issue
4. *Objections*: community members may also Object to a Proposal if they think it conflicts with an OVAL Design Principle
5. *Consensus Building*: the relevant Area Supervisor decides when there is Rough Consensus for adopting a Proposal

6. *Release Process*: the adopted Proposal is released into Development and considered for release into Stable and Official

Create an Issue

An issue can be described as any bug, discussion, new language feature, or update to an existing language feature, which may hold relevance within the OVAL community. When creating an issue within GitHub, a number of potential labels may be applied, based on the context of the issue.

Labels

- **Discussion**: A discussion is meant for general OVAL-related threads. These discussion may reflect implementation details, software updates, and/or suggestions for the OVAL community at large.
- **Question**: An issue labeled "question" should have the intent of being an item which requires resolution by the OVAL community. These issues may consist of implementation-related questions, OVAL language governance-related questions, requests for specific information regarding new schema objects, etc.
- **Bug**: An issue labeled "Bug" represents a technical or implementation-related issue with an OVAL schema or specific construct within a schema. A bug could illustrate to implementers the flaws in a proposed OVAL construct, for example.
- **Website-Docs**: An issue labeled "Website Docs" represents a distinct category of "Bug", specific to "front-facing" OVAL documentation. If process documents require updating, such as how to create an issue, the flaws in those documents should be noted as "website documentation" issues.
- **OVAL-Docs**: An issue labeled "OVAL Docs" represents a distinct category of "Bug", specific to the OVAL Schema documentation. For example, a Test, Object, or State definition contains inadequate explanation, or typo's exist in an element's name or description.
- **Schema-Update**: This issue label is intended to document a proposed update to an already existing OVAL schema. If a contributor would like to add to or remove elements or behaviors from the Unix "file_test" (for example), that proposal would receive this label.
- **Schema-Addition**: This issue label is intended to document a proposed addition to an already existing OVAL schema. A new Microsoft Windows Test/Object/State would be an example of an issue assigned this label.
- **New Schema**: This issue label is intended for those proposals which represent previously non-existing OVAL platform coverage. Perhaps a contributor proposes OVAL language enhancements for Docker, or one of the many cloud platforms (AWS, GCP, Azure).

How To

Creating an issue is very straightforward. Simply navigate a web browser to the OVAL Community and click the "New Issue" button. At this point, the issue creator has options. A number of issue templates exist to assist contributors:

- **Proposal**: This template allows contributors to easily propose additions or changes to the OVAL language.
- **Alternate Proposal**: This template allows contributors to offer an alternative to an existing proposal.
- **Objection**: This template allows contributors to object to, or raise issues regarding an existing proposal.
- **Bug**: This template allows contributors to log an issue with existing OVAL language elements.

Clicking the "Get started" button adjacent to the issue template navigates the user to a pre-filled issue. The contributor may then "fill in the blanks" in the issue template to create the specific ticket.

If none of the issue templates are applicable to the type of issue being created, the contributor can click the "Open a regular issue" link. Add an appropriate title and issue information, describing the issue or question as clearly and succinctly as possible. Select the appropriate issue label(s) by clicking the small "gear" icon adjacent to the "Labels" indicator on the right-hand-side of the page, and submit the issue to the community by clicking the "Submit new issue" button.

FAQs

- Unsure of the label to choose? Select the label(s) which are most relevant to the issue being created. If changes are necessary, the community moderator(s) will adjust the label accordingly.

Initial Proposal

An addition, modification, or removal of OVAL elements begins with an issue. An issue may define an abstract problem statement and recommend one to many language changes. Language contributors may choose to begin the process of proposal and implementation based on an open issue. Starting with the abstract/problem statement in the issue, an initial proposal can be written, expanding on the problem statement, and suggesting potential solutions.

Abstract

The initial proposal submission MUST include an abstract, describing the problem and expanding on the current limitations of OVAL which inform the proposal. Following the abstract, technical documentation SHOULD be included, describing a potential solution.

Technical Documentation

Within the technical documentation, contributors MAY include:

- links to updated OVAL language schema files,
- links to new OVAL schema files,
- descriptive information for any elements being added to OVAL
  - Any new or updated OVAL elements MUST include name, datatype, and description
  - Whenever possible, mappings to underlying system structures or command output SHOULD be included in element descriptions.
- schema or schematron validations.

Finally, the technical documentation SHOULD include proposed collection methods for any new/updated schema elements. For example, if a new test/object/state is being added to OVAL which require parsing of command-line output, the appropriate commands and command options SHOULD be included, potentially including links to relevant "man" pages or any required API documentation.

(This may change - Based on the subsequent modification/acceptance of the proposal, workflow may dictate that sample content/results should come later)

Following the technical documentation, contributors MAY include sample content, from which implementers can begin the review and consensus process. If the proposal author(s) have already implemented the functionality represented in the proposal, sample content plus OVAL results MAY be included.

How To

The initial proposal process MUST begin with a pull request.

- The title of the pull request MUST follow the `Proposal: <Title>` naming convention
- The labels for the pull request MUST
  - correspond to the label assigned to the issue for which this proposal was initiated, as noted in the create an issue section, and
  - contain any `<Area: area-name>` labels as needed.
- The body of the pull request MUST include a reference to the issue for which this proposal was initiated.

Submission of the pull request affirms the intent to begin the consensus building process, and to begin the 45-day comment period.

Alternate Proposals

Brief description and purpose.

How To

Step-by-step instuctions including CLI samples if appropriate.

FAQs

Some FAQs about steps and any associated process details.

Documentation Links

Links to process docs?

Objections

Objections are a mechanism for blocking an open Proposal that conflicts with an *OVAL Design Principle*.

First Steps

Before making an Objection:

1. **Collaborate**: You MUST make a good faith attempt to engage with the Proposal Author and wider community by commenting on the Proposal (i.e. the Pull Request) and explaining your concerns. Typically, the community can work together on the Proposal to clarify any misconceptions and revise the Proposal, if necessary, to avoid any negative impact.
2. **Allow Time**: You MUST give the Proposal Author and community some time (at least a few days) to respond to your comments.
3. **Suggest a Better Way**: If possible, you SHOULD make an Alternate Proposal to resolve the Issue and address your concerns.

How To

1. Complete the *First Steps* above
2. Identify the specific *OVAL Design Principles* that the Proposal conflicts with. Or, if the Proposal does not conflict with an existing Design Principle, you may draft and propose a new Design Principle.
3. Create a New Issue Using the Objection Template.
4. Add a comment to the Proposal PR linking to the Objection Issue
5. The relevant Area Supervisor will give the community a reasonable amount of time (at least a few days) to weigh in on the Objection and then, when appropriate in their judgement, *assess consensus*.
   - If the Objection is accepted, the Supervisor will:
     - Add a comment to the Proposal PR indicating that the Objection was accepted
     - Reject the Proposal PR
     - Add a comment to the Objection Issue indicating that it was accepted
     - Close the Objection Issue
   - If the Objection is not accepted, the Supervisor will:
     - Add a comment to the Proposal PR indicating that the Objection was not accepted
     - Add a comment to the Objection Issue indicating that it was not accepted
     - Close the Objection Issue

FAQs

What if my Objection is not accepted and the Proposal is accepted? You may follow the Proposal Process to further improve OVAL.

Consensus Building

The intent behind consensus in the context of the OVAL Community is the notion of what is known as rough consensus. This does not require all participants to agree, and it's not a voting mechanism. The gist of it is that no serious, unaddressed objections exist. As mentioned in the previously described characteristics, it will be up to the Area Supervisors to judge rough consensus. Then, if there is an objection to a proposal, and that objection has been sufficiently addressed, then we can move forward. Conversely, if there is an objection that has not been sufficiently addressed, we know the process cannot move forward. This notion has worked well in other venues, and we believe it will serve us well here also.

How To

The consensus building process involves four roles: Proposer, Objector, Area Supervisor, Leadership Board. Not all roles are involved in each instance of conensus building.

Initial Proposal

The consensus process begins when a Proposer makes a proposal (see Initial Proposal). At this point, the 45-day clock is started - if no objections/issues are raised within this 45-day period, the proposal moves directly into Consensus

Call, as described below. From time to time, an initial proposal may be a response to another proposal for which one or more issues have been raised.

Objection Handling

After an initial proposal, one or more issues may be raised by anyone in the community. We refer to anyone raising such an issue as an Objector, even though some objections may be be, for example, a clarifying question. When an issue is raised, it will fall into one of the following categories:

- **Clarifying question:** The Proposer must provide an answer to the question within a reasonable amount of time.
- **Objection based on existing, or presumed missing, Design Principle:** Sometimes a proposal may be in violation of an existing Design Principle. When this is the case, the proper response is to simply address the objection as such. From time to time we expect to update our Design Principles based on an objection. When this is the case, the Proposer, Objector, and Area Supervisor work to update the Design Principle according to this process.
- **Objection based on some other reason (i.e. unnecessary or technically unsound proposal):** The Proposer works to reasonably address the objection.

In each of the above scenarios, an entire discussion may result, and at some point there will be a conclusion to the discussion. From time to time the conclusion might be a perceived impasse. Additionally, an *alternate proposal* may be created (see Alternate Proposals).

When all issues and resulting discussions are concluded, then the process moves to into Consensus Call.

Consensus Call

The purpose of a Consensus Call is to provide a 14-day period during which community members who may have missed the initial proposal and subsequent discussions have an opportunity to opine. The Area Supervisor determines when an effort is ready for a Consensus Call. From time to time, the Leadership Board may be consulted when a consensus call is too difficult for an Area Supervisor to judge.

Once consensus has been reached (see About Consensus below) as judged by the Area Supervisor and the 14-day Consensus Call has come to a close, the proposal moves to release (see Release Process).

About Consensus

Because our process is open to the possibility of accepting proposals that may not enjoy wide interest, it may end up being the case that a proposal achieves rough consensus, even though only two or three parties support it (everyone else may be completely indifferent). In this situation, there are no objections to the proposal, but there's also not necessarily a critical mass of support. In this case, there is no harm in such a minimally supported proposal "making it through" - there are, after all, no objections to the proposal, and we must assume that those interested in that particular area are paying attention enough to otherwise object. In other words, this is why we have the desire to retain and keep interested and active Area Supervisors.

Reaching consensus is really about carrying on a conversation. Because each proposal is composed of a GitHub issue and pull request, such conversation should be conducted via those GitHub constructs, so that the entire community sees the conversation as it unfolds, and therefore has an opportunity to opine, should they have an opinion worth stating.

It is the responsibility of all interested parties to achieve consensus. Getting to consensus is not necessarily the Area Supervisor's responsibility, nor is it exclusively the proposer's responsibility. Here are some guidelines that may help in reaching consensus:

- Ensure the proposal is clearly scoped

- List each concern participants have about the proposal
- Use quick, simple polls to quickly guage interest in a solution[1]
- Understand that lack of disagreement is *more important* than total agreement
- Rough consensus is achieved when all issues are addressed, but not necessarily accommodated
- Rough consensus is *not* about for and against cohorts

FAQs

What are the principles of consensus building?

- **Inclusion:** Everyone in the community has a voice, and their voice is valued.
- **Participation:** People are participating - consensus is not nearly as effective when only a few (or worse, no one but the proposer) are really involved int he discussion.
- **Cooperation:** Individuals and organizations need to work toward the common goal of finding a solution in the best interest of the community.
- **Egalitarianism:** No one person's voice necessarily carries any more weight than an other's.
- **Solution-mindedness:** Always keep the solution in mind, which helps to avoid any perceived inter-organizational/-personal conflicts

What are the benefits of using a consensus processs?

An ideal outcome of a consensus process is that everyone is enthusiastically supportive of a proposal. However, the well-known aphroism states, "perfect is the enemy of the good". The real benefit of a consensus process is that parties with sometimes differing perspectives and needs are satisfied, if not necessarily emhpatically pleased, with a given solution.

Does consenting to a solution mean it's my first choice?

Not at all. Consenting to a solution simply means that you agree to the solution being proposed, not that it's your ideal solution. Sometimes this is known as disagree and commit. You may disagree that this is the *best* solution, from your singular perspective, but commit to supporting the solution as the *overall better* solution for the community as a whole.

Who judges rough consensus?

As described in our process, the Area Supervisor (from time to time under the guidance of the Leadership Board) will be the judge of rough consensus.

What about this for and against cohort thing?

A really good treatment is found in RFC7282 of the IETF. There are two sections in particular that describe situations where there may be vast numbers for or against, but the rough consensus is still against or for respectively. In cases like this, the Area Supervisor's challenge will be to sift through the yeas and the nays to determine which of those voices have been *active throughout the discussion regarding the proposal*.

Documentation Links

- An IETF Informational document on rough consensus

[1] Doodle Polls (see this) are a good way to conduct simple polls.

Release Process

The OVAL release process will see a given proposal being, in effect, accepted into one of four release streams:

1. **Extensions:** Proposals that are made available as an *unofficial* extension after not being adopted by the community.
2. **Development:** All adopted proposals awaiting a sufficient number of qualifying-implementations or the next scheduled stable release date.
3. **Stable:** All adopted proposals that have a sufficient number of Qualifying Implementations as of the most recent stable release date on February 1st and August 1st of each year.
4. **Official:** A stable release that the OVAL board selects as the Official release at least once each year.

How To: Release Streams

Extension Stream Release Process

The extension release stream includes proposals that were not adopted, but have been made available as unofficial extensions to the OVAL Language. Extensions enable community members to publicly document schemas that are not adopted by the community.

1. A proposal is not adopted (e.g. fails due to an objection or an alternative proposal being selected).
2. The extension MUST:

1. Be expressed as a valid XSD
2. Have publicly available documentation
3. Have at least one implementation
4. Have at least one, and preferably several, content examples

3. Any community member requests that the proposal is published as an extension by adding a comment to the rejected proposal
4. The relevant Area Supervisor publishes the proposal to the extensions release stream

Extension Stream CLI Example

TBD

Development Stream Release Process

The development release stream includes all adopted proposals that are awaiting sufficient qualifying-implementations or the next scheduled Stable Release date.

1. A proposal is adopted via the standard community process
2. The relevant Area Supervisor publishes the Proposal immediately to the Development release stream

Development Stream CLI Example

TBD

Stable Stream Release Process

The stable release stream includes all proposals that have been adopted by the community via the standard proposal process and have a sufficient number of qualifying-implementations as of the semi-annual stable release dates (February 1 and August 1 of each year).

1. A proposal in the development release stream is implemented by a sufficient number of qualifying-implementations
2. On a release date, Area Supervisors publish all such proposals to the Stable release stream no later than 23:59 UTC.

Stable Stream CLI Example

TBD

Official Stream Release Process

The official release stream is a stable release selected by the OVAL Board at least once per year to be the official release.

1. The OVAL Board selects the official release via vote at a regular meeting

Official Stream CLI Example

TBD

FAQs

No. While having a Qualified Implementation is preferable (see qualifying-implementations), an extension implementation does not need to be a Qualifying Implementation and may simply be a proof of concept.

- The `master` is the official release stream, and will be labeled with the current release versions appropriately.
- The `stable` branch represents the stable release stream
- The `development` branch represents the development release stream
- The `extension` branch represents the extensions release stream

Documentation Links

Links to process docs?

5.7 Developer Guides

5.7.1 Editing These Guidelines

Are you interested in contributing a change to these guidelines? If so, thank you!

This document is intended to help you get up and running.

Prerequisites

You'll need the following in order to get started:

1. A fork of The OVAL Community github repository
2. A local working copy of your fork
3. Python 3.x

Initial Setup

In order to build the guidelines locally and review your changes, you'll need to install a few Python modules:

```
# cd to the root of your fork of this repository
cd OVAL

# create a python virtual environment (recommended) and activate it
python3 -m venv ./venv
. ./venv/bin/activate

# install required python modules
pip install -r tools/requirements.txt
```

Building the Guidelines

On Windows:

```
cd guidelines
make.bat
```

On Mac/Linux:

```
cd guidelines
make html
```

Using Sphinx Auto Build/Reload (Windows/Max/Linux):

```
cd guidelines
sphinx-autobuild . ./_build/html
```

Using reStructuredText

These guidelines are written in reStructuredText. Learn more here:

- Sphinx reStructuredText Primer
- Offical reStructuredText Documentation

Important Guideline Guidelines

These guidelines use the following section heading formats:

```
Page Header
===========

Section Header
--------------

Subsection Header
^^^^^^^^^^^^^^^^^

SubSubsection Header
""""""""""""""""""""
```

When updating these guidelines, please note the following:

- Every page MUST start with a Page Header (underlines with "=") and MUST only contain 1 Page Header

These guides are intended to help community members learn the mechanics of contributing to the community.

- *Editing These Guidelines*

5.8 Release Streams

The OVAL Community maintains 4 different OVAL Language release streams:

- *Development*: all adopted Proposals awaiting release into Stable
- *Stable*: all adopted Proposals qualified for release into Stable as the most recent February 1st or August 1st
- *Official*: a Stable Release that the OVAL Board selects as the Official release at least once a year
- *Extensions*: Proposals that were not adopted, but are made available as Unofficial Extensions

For more details on the release process, see *Release Process*.

5.8.1 Development

Once a Proposal is Adopted by the Community, it is immediately published into the Development release stream where it will stay until it has sufficient *Qualifying Implmentations* and is released into *Stable*.

5.8.2 Stable

On February 1st and August 1st of each year, all Proposals that have been Adopted by the Community and have a sufficient *Qualifying Implmentations* are released into *Stable*.

5.8.3 Official

The Official release stream is a Stable release selected by the OVAL Board at least once per year to be the Official release.

5.8.4 Extensions

The Extensions release stream includes Proposals that were not Adopted, but have been made available as unofficial extensions to the OVAL language at the request of a Community member. Extensions enable Community members to publicly document schemas that are not adopted by the Community.

5.9 OVAL Versioning

Details of OVAL core and platform versioning schemes ported from:

- http://ovalproject.github.io/documentation/policy/versioning/
- https://github.com/OVALProject/Language/issues/281
- http://ovalproject.github.io/documentation/policy/deprecation/

5.10 OVAL Content Repositories

OVAL Content Repositories exist to preserve OVAL definitions and make them available for public use. While the CIS Repository is the official repository, others exist that contain content specific to operating systems, applications, and software vendors.

5.10.1 Repositories

Below is a list of additional known OVAL repositories.

ALTEX SOFT

The ALTEX-SOFT repository OVALdb consist of OVAL Definitions that correspond to security advisories/notices/bulletins/compliances for a lot of software vendors. This repository contains OVAL definitions for vulnerabilities, patches, compliances and inventories.

Click here for the ALTEX-SOFT Repository

CANONICAL

This page provides OVAL xml content for the latest Ubuntu operating system versions.

Click here for the Ubuntu Repository

CIS™ Center for Internet Security®

The CIS repository is the new official OVAL Repository following the transition away from MITRE. Created August 2015.

Click here for the CIS Repository

The Cisco Security Intelligence Operations repository consists of Cisco security advisories in the standardized Common Vulnerability Reporting Format (CVRF) and includes OVAL Vulnerability Definitions for the Cisco IOS security advisories. Created September 2012.

Click here for the Cisco Repository

The Debian repository of OVAL content consists of OVAL Definitions that correspond to Debian security advisories. Created August 2010.

Click here for the Debian Repository

Defense Information Systems Agency Field Security Operations (DISA FSO)

A repository of Security Technical Implementation Guides (STIGs) in support of Security Content Automation Protocol (SCAP) content and tools. Created: May 2012.

Click here for the DISA FSO Repository

IT Security Database

This site collects OVAL Definitions from sources such as the OVAL Repository, Red Hat, Suse, NVD, Apache, etc., and provides a unified, easy-to-use Web interface to all IT security related items about them including patches, vulnerabilities, and compliance checklists. Created: November 2010.

Click here for the IT Security Database Repository

The Security Content Automation Program (SCAP) is a public free repository of security content to be used for automating technical control compliance activities, vulnerability checking (both application misconfigurations and software flaws), and security measurement. Created January 2007.

Click here for the NIST (SCAP) Repository

The Positive Technologies repository of OVAL content consists of OVAL Definitions collected from various sources. Created May 2012. **Note that this repository is currently inactive but is being reworked as of October 2018.**

Click here for the Positive Technologies Repository

The Red Hat repository of OVAL content consists of OVAL Patch Definitions that correspond to Red Hat Errata security advisories. Created May 2006.

Click here for the Red Hat Repository

SecPod SCAP Feed, also hosted as a repository, is a service providing standardized SCAP content (CVE™, CPE™, CCE™, XCCDF, and OVAL®) for vulnerability, patch, inventory, and compliance management. Created December 2010.

Click here for the SecPod Repository

SECURITY DATABASE
KEEP A VIGILANT EYE ON YOUR DEFENSES

This Web site provides a mirror of the OVAL Repository and links its Alerts to OVAL Definitions when possible. Created February 2012.

Click here for the Security Database Repository

The SUSE Linux Enterprise OVAL Information database is an index of fixed security incidents indexed by product, RPM package name and version for use in security compliance checking. Created July 2010.

Click here for the SUSE Repository

5.10.2 Updating the List

To make changes to this list; to be added, removed, or have existing content modified,

5.11 OVAL Support Declarations

Intro tbd.

5.11.1 Declarations

Altex-Soft

Declared January 30, 2012
www.altex-soft.com
www.altx-soft.ru

> Altex-Soft OVALdb
> *Web-Based OVAL Repository Database*

[ ] Authoring Tool
[ ] Definition Evaluator
[X] Definition Repository
[ ] Results Consumer
[ ] System Characteristics Producer

RedCheck
Vulnerability, Patch, and Compliance Assessment

[ ] Authoring Tool
[X] Definition Evaluator
[ ] Definition Repository
[ ] Results Consumer
[ ] System Characteristics Producer

Beyond Security

Declared August 7, 2013
www.beyondsecurity.com

The beSECURE family of network Vulnerability Assessment and Web Application Security testing solutions are the most accurate and easiest to use in the industry. beSECURE uses OVAL to import benchmarks from the OVAL repository and user-developed XML files and to export assessment results files. beSECURE is available as a network appliance or hosted solution and will deliver layer 3-7 scanning to businesses and government units of any size. It will find, prioritize and manage the repair of security weaknesses in your network and web applications with the fastest setup and the least maintenance possible.

beSECURE
Vulnerability and Configuration Assessment and Management

[ ] Authoring Tool
[X] Definition Evaluator
[ ] Definition Repository
[X] Results Consumer
[X] System Characteristics Producer

BeyondTrust

Declared September 8, 2010
www.beyondtrust.com

BeyondTrust is an innovative leader in vulnerability and security research, providing security solutions that help businesses and users protect their systems and intellectual property from compromise.

Retina Network Security Scanner

Vulnerability Assessment

[ ] Authoring Tool
[X] Definition Evaluator
[ ] Definition Repository
[ ] Results Consumer
[X] System Characteristics Producer

Center for Internet Security

Declared February 26, 2014
www.cisecurity.org

CIS-CAT is an SCAP-compliant, host-based configuration assessment tool primarily designed to perform compliance assessments against recommendations contained in CIS benchmarks. OVAL-based compliance content developed by third parties, such as DISA and NIST, is also supported by CIS-CAT for major Microsoft products, including Windows, Office, Internet Explorer, and SQL server, as well as Red Hat Enterprise Linux platforms. CIS-CAT's support for OVAL also affords users the ability to perform compliance, vulnerability, inventory, and patch assessments using content generated from numerous sources, including CIS, DISA, and NIST/USGCB, from a single tool.

Center for Internet Security Configuration Assessment Tool (CIS-CAT)
Host-Based Configuration Assessment Tool

[ ] Authoring Tool
[X] Definition Evaluator
[ ] Definition Repository
[ ] Results Consumer
[X] System Characteristics Producer

Cisco Systems, Inc.

Declared February 10, 2012
www.cisco.com

Traditionally, Cisco discloses information required for an end-user to assess the impact of a vulnerability and any potential steps needed to protect their environment. This information includes all the required technical information for customers to ascertain appropriate remedial action. OVAL provides a framework that allows vendors and their customer to determine if a software vulnerability or patch exists on a given system. Cisco is in the process of adopting OVAL for vulnerability disclosure. Cisco IOS security vulnerability OVAL content is currently supported. Additional products are being considered in the future.

Cisco PSIRT Security Advisories and Vulnerability Disclosures
Cisco Repository of OVAL Content

[ ] Authoring Tool

[ ] Definition Evaluator
[X] Definition Repository
[ ] Results Consumer
[ ] System Characteristics Producer

Defense Information Systems Agency Field Security Operations (DISA FSO)

Declared July 18, 2012
iase.disa.mil/stigs/

DISA is adopting OVAL for leveraging enterprise compliance and vulnerability assessment for the U.S. Department of Defense (DoD). Utilizing COTS-based scan engines, DISA is transforming security requirements from prose base documents to machine readable content. This content utilizes the OVAL Language as a mechanism to determine results for secure net worthiness in the DoD while supporting the war fighter.

DoD SCAP Content Repository
SCAP Content Repository

[ ] Authoring Tool
[ ] Definition Evaluator
[X] Definition Repository
[ ] Results Consumer
[ ] System Characteristics Producer

Information-Technology Promotion Agency

Declared January 30, 2012
www.altex-soft.com
www.altx-soft.ru]

IPA offers two products for JVN Security Content Automation Framework. Version Checker is an OVAL-based, free, easy-to-use scanner that allows people to easily check whether the software installed on their PC is the latest version. With just one mouse click, people can check the versions of multiple software. The results are easy to understand: a tick mark signifies the latest version and a cross mark signifies an obsolete version. If the software is not the latest version, users can easily access the vendor's download website with just a few clicks. MyJVN API is a software interface to access and utilize vulnerability countermeasure information and OVAL repository stored in JVN and JVN iPedia. To enable application developers to use data through an open interface, JVN iPedia has adopted SCAP, a set of standards for describing vulnerability countermeasure information.

MyJVN API
Vulnerability Assessment and Configuration Management

[ ] Authoring Tool
[ ] Definition Evaluator
[X] Definition Repository

[ ] Results Consumer
[ ] System Characteristics Producer

MyJVN Version Checker
Vulnerability Assessment

[ ] Authoring Tool
[X] Definition Evaluator
[ ] Definition Repository
[ ] Results Consumer
[ ] System Characteristics Producer

Institute for Information Industry

Declared December 12, 2012
www.iii.org.tw

CSK controller performs automatic compliance auditing to each CSK agent on enterprise endpoints. It can check security misconfigurations, scan systems and application vulnerabilities, evaluate enterprise threats through the baselines which is in the context of XCCDF based on enterprise demands or official compliance. CSK agent gathers all the security information including system configurations, application weakness, service status on each endpoint. Moreover, CSK agent also sends the security content according to the OVAL and CCE definitions to the controller for generating the human-readable reports evaluated by CVSS and specified baselines (USGCB, MS-baselines).

Crystal Security Keeper
Vulnerability Assessment, Configuration Management, Auditing and Centralized Audit Validation

[ ] Authoring Tool
[X] Definition Evaluator
[X] Definition Repository
[P] Results Consumer
[ ] System Characteristics Producer

Joval

Declared February 26, 2014
www.jovalcm.com

Joval Continuous Monitoring
Open Source, Java-based OVAL Definition Interpreter

[ ] Authoring Tool
[X] Definition Evaluator
[ ] Definition Repository
[ ] Results Consumer

[X] System Characteristics Producer

Nakamura Akihito

Declared January 14, 2011
github.com/nakamura5akihito
formerly under AIST at www.aist.go.jp

SIX OVAL is a free and open-source Java class library to build enterprise compliance/vulnerability management applications. The main parts are OVAL domain model and object-XML/object-RDB data mapping. It also provides off-the-shelf server/client components including a repository of definitions and results at the central server, which can be searched from and posted to via a web service connection from any number of clients. The client is capable of getting definitions from the repository, evaluating the content on the local host, and reporting the results back to the central server.

SIX OVAL
Enterprise Compliance/Vulnerability Management

[ ] Authoring Tool
[X] Definition Evaluator
[ ] Definition Repository
[ ] Results Consumer
[ ] System Characteristics Producer

New Net Technologies, Ltd.

Declared May 30, 2014
www.nntws.com

NNT Change Tracker Enterprise provides continuous protection against known and emerging cyber security threats in an easy to use solution. NNT Change Tracker leverages OVAL Definitions to provide vulnerability and compliance assessments for a wide-range of platforms and devices. Options provided for both agent-based and agentless vulnerability scans of a wide range of database systems, operating systems, appliances and network devices. NNT Change Tracker is also a CIS Certified Vendor Product for CIS Benchmark Checklist validation.

NNT Change Tracker
Vulnerability and Compliance Assessment and Management, Host-Based Intrusion Detection

[ ] Authoring Tool
[X] Definition Evaluator
[P] Definition Repository
[P] Results Consumer
[X] System Characteristics Producer

OpenVAS

Declared July 6, 2012
www.openvas.org/

OpenVAS is a vulnerability management and vulnerability scanning software framework. A feed service allows regular updates of Network Vulnerability Tests (NVTs). The main security scan phase of the application collects security information about each host in the network being scanned. Subsequently, comprehensive OVAL-related processing is possible. This includes exporting system characteristics for the whole network, and applying the applications reporting framework according to OVAL Definitions.

> OpenVAS
> *Vulnerability Management*
>
> [ ] Authoring Tool
> [P] Definition Evaluator
> [ ] Definition Repository
> [P] Results Consumer
> [X] System Characteristics Producer

Red Hat, Inc.

Declared February 10, 2010
www.redhat.com

Red Hat was a founding board member of the OVAL project and has been publishing OVAL Vulnerability Definitions for Red Hat Enterprise Linux Security Advisories since 2006. This initiative forms part of our commitment to make the deployment of security ubiquitous through the use of industry-wide standards.

> Red Hat Security Advisories
> *OVAL Definition Repository*
>
> [ ] Authoring Tool
> [ ] Definition Evaluator
> [X] Definition Repository
> [ ] Results Consumer
> [ ] System Characteristics Producer

Resolver

Declared February 26, 2014
www.resolver.com

In order to promote open standards and leveraging existing tools already deployed as authoritative sources of risk, threat, security, governance, and compliance audit details, Resolver's big data risk management software platform, Resolver RiskVision, consumes OVAL Definitions, OVAL Results, and OVAL System Characteristics via its user interface or via data connectors. As a consumer of OVAL attributes, RiskVision supports OVAL 5.10.1 and prior versions. In addition, RiskVision accommodates SCAP in its 'XCCDF and OVAL' import tool.

Resolver RiskVision
Big Data Risk Management Software

[ ] Authoring Tool
[ ] Definition Evaluator
[ ] Definition Repository
[X] Results Consumer
[ ] System Characteristics Producer

SecPod Technologies

Declared December 10, 2010
www.secpod.com

SecPod is an information security research and development company offering services in the area of threat detection and management. SecPod supports OVAL, an open standard to provide security automation. SecPod SCAP Feed is a service providing Vulnerability, Inventory, Compliance, and Patch definitions covering majority of the CVE's for various operating systems, enterprise servers, and applications. The feed, also hosted as a repository, is backed with professional support, can be integrated into vendor products, and also consumed by end users. SecPod Saner is a light-weight, easy-to-use enterprise grade vulnerability mitigation software that proactively assesses and secures endpoint systems. SecPod Saner adopts OVAL natively consuming the SCAP feed from the SecPod SCAP Repo content repository.

SecPod SCAP Feed
OVAL Repository

[ ] Authoring Tool
[ ] Definition Evaluator
[X] Definition Repository
[ ] Results Consumer
[ ] System Characteristics Producer

SecPod Saner
Vulnerability Management

[ ] Authoring Tool
[X] Definition Evaluator
[ ] Definition Repository
[ ] Results Consumer
[X] System Characteristics Producer

SPAWAR Systems Center Atlantic

Declared February 25, 2010
www.public.navy.mil/spawar/Atlantic/

The SCAP Compliance Checker has adopted OVAL as part of the FDCC Scanner capabilities of SCAP Validation Program. SCAP Compliance Checker is able to process all four of OVAL's schemas: the Definitions schema, the System Characteristics schema, the Results schema and the Variables schema. SCAP Compliance Checker processes the XCCDF content of a SCAP stream and extracts any variables that need to be imported into the OVAL engine. It then creates an XML file using the OVAL Variables schema that contains these variables. The OVAL engine later uses this file during OVAL processing. By using the industry standard OVAL schemas, SCAP Compliance Checker can share data with any tool that understands OVAL.

> SCAP Compliance Checker
> *OVAL Definition Evaluator*
>
> [ ] Authoring Tool
> [X] Definition Evaluator
> [ ] Definition Repository
> [ ] Results Consumer
> [X] System Characteristics Producer

SUSE

Declared February 28, 2014
www.secpod.com

Our customers need an index of fixed security incidents indexed by product, RPM package name, and version for use in their security compliance checking. As they are using a wide range of checking tools inventing a new format would have caused unnecessary work on all sides. We have chosen to use the OVAL format for publishing this data, which is in our eyes the accepted industry standard format for this purpose.

> SUSE Linux Enterprise OVAL Information
> *Database*
>
> [ ] Authoring Tool
> [ ] Definition Evaluator
> [X] Definition Repository
> [ ] Results Consumer
> [ ] System Characteristics Producer
>
> SUSE Manager 1.7
> *Linux Patch and Configuration Management*
>
> [ ] Authoring Tool
> [X] Definition Evaluator
> [ ] Definition Repository

[X] Results Consumer
[X] System Characteristics Producer

ToolsWatch

Declared April 14, 2015
http://www.toolswatch.org/

SSA (Security System Analyzer) is free non-intrusive OVAL/XCCDF host-based security analyzer and compliance tool. It introduces a new simplified way to rely on open standards such OVAL and XCCDF to report compliance issues. SSA has adopted the OVAL standard as part of its vulnerability validation process. As a result, SSA consumes the Definitions and solely relies on the OVAL and XCCDF interpreters. vFeed provides a full aggregated, cross-linked and standardized Vulnerability Database based on CVE and standards such as OVAL, CPE, CWE, CAPEC, CVSS etc. Therefore, it introduces a new simplified XML format that expands the vulnerability coverage and correlation around the CVE. vFeed has adopted the OVAL as part of its correlation and aggregation capability. As a result, vFeed consumes the OVAL XML definitions, extract and map variables to expand the CVEs data.

SSA - Security System Analyzer
Security Scanner and Compliance Assessment Software

[X] Authoring Tool
[X] Definition Evaluator
[ ] Definition Repository
[X] Results Consumer
[X] System Characteristics Producer

vFeed API and Vulnerability Database Community
Vulnerability and Threats Database

[ ] Authoring Tool
[ ] Definition Evaluator
[X] Definition Repository
[X] Results Consumer
[ ] System Characteristics Producer

Tripwire, Inc.

Declared October 19, 2010
http://www.tripwire.com/

Tripwire provides a comprehensive suite of file integrity, policy compliance, and log and event management solutions. Tripwire Enterprise automates change detection and misconfiguration correction to reduce risk of exploits and breaches. Tripwire Enterprise provides SCAP functionality that includes the ability to process OVAL content.

Tripwire Enterprise

Security Configuration Management

[ ] Authoring Tool
[X] Definition Evaluator
[ ] Definition Repository
[X] Results Consumer
[X] System Characteristics Producer

VMware

http://www.vmware.com/

Enhanced SCAP Content Editor
OVAL Authoring Tool

[X] Authoring Tool
[ ] Definition Evaluator
[ ] Definition Repository
[ ] Results Consumer
[ ] System Characteristics Producer

5.11.2 Updating the List

To add to, remove from, or edit this list, please submit a pull request.

5.12 OVAL Mailing Lists

OVAL_Repository A list intended to be used for CIS OVAL Repository discussions, including discussions pertaining to repository submissions, and the subject of those submissions (i.e. vulnerabilities, configurations, inventory, and so on).

OVAL_Developer A list used to discuss OVAL as a language and to provide support to its implementation community.

OVAL_Board A list intended to be used by members of the OVAL Board for Board-related discussions. Participation on this list is by invitation.

5.13 Additional Resources

The following resources are related standards, OVAL documentation, and other helpful information. Suggested additions or deletions can be submitted through the developer mailing list.

5.13.1 Resources

- OVAL Community Repository *The official repository for OVAL development, maintenance, issue tracking, enhancement proposals, etc.*
- OVAL Developer Guide *The OVAL Developer Guide was written by MITRE, the previous OVAL Sponsor.*
- Security Content Automation Protocol (SCAP) home *SCAP is a set of standards defined by NIST that includes OVAL.*
- Center for Internet Security *The website for the current OVAL Sponsor.*
- OVAL Repository *A publicly contributed/available repository of OVAL definitions of varying use-cases, OS platforms, and product families.*

5.13.2 Related Standards

- The eXtensible Configuration Checklist Description Format *XCCDF defines a structured collection of security configuration rules for some set of target systems.*
- Script Check Engine *SCE defines a check system allowing XCCDF rules to evaluate compliance based on the results of execution of scripts in various scripting languages.*

5.13.3 Archive

Historical and archived information regarding previous versions of OVAL, prior sponsor documentation, etc. may be found at the following resources:

- MITRE OVAL Homepage *Deprecated website containing historical OVAL data.*
- MITRE OVAL Archives *The OVAL Language archives.*
- OVAL Mailing List Archives *An archive of OVAL-related mailing list discussions.*
- OVAL Language Schemas *Up to and including OVAL v5.11.2, the OVAL Language GitHub repository provides the official OVAL language documentation, schemas, and tools used in the development of OVAL interpreters.*
- OVAL Language Sandbox *The OVAL Language Sandbox, hosted on GitHub.com, provides a collaborative environment for the community to propose and develop experimental capabilities for the OVAL Language.*
- NIST XCCDF Homepage *XCCDF is a specification language for writing security checklists, benchmarks, and related kinds of documents.*

5.14 Terms of Use

5.14.1 Introduction

OVAL is an open standard developed by the information security community as represented by the OVAL Board and the OVAL discussion lists, and maintained by the Center for Internet Security, Inc. under license from the United States Department of Homeland Security (U.S. DHS) on this public OVAL Website.

The OVAL Language and any resulting OVAL content based upon the language that is stored in the OVAL Repository are free to use by any organization or individual for any research, development, and/or commercial purposes, per below.

The United States Government has copyrighted the OVAL Language for the benefit of the community in order to ensure it remains a free and open standard, as well as to legally protect the ongoing use of it and any resulting content by government, vendors, and/or users. The United States Government has trademarked ® the OVAL acronym and the OVAL logo to protect its sole and ongoing use by the OVAL effort within the information security arena.

Please submit through the developer mailing list, if you require further clarification on this issue.

5.14.2 Open Vulnerability and Assessment Language (OVAL®) License

Your use of OVAL is conditioned upon acceptance of the following terms and conditions:

OVAL is comprised of the OVAL Language, OVAL Content, and the OVAL Repository. Each is defined below:

- The OVAL Language serves as the framework and vocabulary of OVAL. The Language covers the three steps of the assessment process: an OVAL System Characteristics schema for representing system information, an OVAL Definition schema for expressing a specific machine state, and an OVAL Results schema for reporting the results of an assessment.
- OVAL Content is content written in the OVAL Language. All content written in the OVAL Language is considered OVAL Content.
- The OVAL Repository is a collection of OVAL Content hosted by the Center for Internet Security, Inc. under license from the U.S. DHS. It is the central meeting place for the OVAL Community to discuss, analyze, store, and disseminate OVAL Definitions. Each definition in the OVAL Repository determines whether a specified software vulnerability, configuration issue, program, or patch is present on a system.

The Center for Internet Security, Inc. (CIS) under license from U.S. DHS hereby grants you a non-exclusive, royalty-free, worldwide license to use OVAL for research, development, and commercial purposes. Any copy you make for such purposes is authorized provided that you reproduce the United States Government's copyright designation and this license in any such copy.

The OVAL Language is the copyrighted work of the United States Government. No ownership or other proprietary interest in the OVAL Language is granted to you other than what is granted in this license.

The names and trademarks for OVAL may not be used in association with commercial products. Notwithstanding the foregoing, commercial products that are based upon or incorporate any portion of OVAL may use a word mark as part of a factual statement that references the commercial products' use of OVAL materials, but only in a manner that does not imply DHS's endorsement of the commercial product.

OVAL Content, whether already in the OVAL Repository hosted by CIS under license from the U.S. DHS or developed by you and sent to CIS via the discussion forums or any other means to be deposited into the OVAL Repository, is fully available for public use free of charge. In addition, to the extent that contributed OVAL Content involves pre-existing copyrighted works, you hereby grant to CIS and the United States Government an irrevocable, worldwide, royalty-free, non-exclusive, license, for the duration of the copyright, to do the following:

- to reproduce such OVAL Content, either alone or as part of a collective work;
- to translate, adapt, alter, transform, modify, or arrange such OVAL Content, thereby creating derivative works ("Derivative Works"); and
- to distribute, display, or communicate copies of such OVAL Content to the public free of charge.

ALL DOCUMENTS AND THE INFORMATION CONTAINED THEREIN ARE PROVIDED ON AN "AS IS" BASIS AND THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE CENTER FOR INTERNET SECURITY, INC., ITS DIRECTORS, OFFICERS, EMPLOYEES CONTRACTORS, AND AGENTS, AND THE UNITED STATES GOVERNMENT 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.

You understand that any export of the OVAL Language, OVAL Content, or OVAL Repository may require an export license and you assume full responsibility for obtaining such license.

This License shall be construed, governed, interpreted and applied in accordance with the laws of the State of New York without regard to any conflict of law rules and you agree to submit to the exclusive jurisdiction of the State of New York courts.