

# - Open Vulnerability and Assessment Language - Element Dictionary

- Schema: IOS Definition
- Version: 5.0 release candidate 3
- Release Date: 26 May 2006

The following is a description of the elements, types, and attributes that compose the 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 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>.

---

---

## < 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. The evaluation of the test is guided by the check attribute that is inherited from the TestType.

Child Elements	Type	MinOccurs	MaxOccurs
object	oval-def:ObjectRefType	1	1
state	oval-def:StateRefType	0	1

## < 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 comparison. 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.

Child Elements	Type	MinOccurs	MaxOccurs

global_command	oval-def:EntityObjectStringType	1	1
----------------	---------------------------------	---	---

### < 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.

Child Elements	Type	MinOccurs	MaxOccurs
global_command	oval-def:EntityStateStringType	1	1

### < interface\_test >

Child Elements	Type	MinOccurs	MaxOccurs
object	oval-def:ObjectRefType	1	1
state	oval-def:StateRefType	0	1

### < interface\_object >

Child Elements	Type	MinOccurs	MaxOccurs
name	oval-def:EntityObjectStringType	1	1

### < interface\_state >

Child Elements	Type	MinOccurs	MaxOccurs
name	oval-def:EntityStateStringType	0	1
ip_directed_broadcast_command	oval-def:EntityStateStringType	0	1
no_ip_directed_broadcast_command	oval-def:EntityStateStringType	0	1
proxy_arp_command	oval-def:EntityStateStringType	0	1
shutdown_command	oval-def:EntityStateStringType	0	1

### < 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. The evaluation of the test is guided by the check attribute that is inherited from the TestType.

Child Elements	Type	MinOccurs	MaxOccurs
object	oval-def:ObjectRefType	1	1
state	oval-def:StateRefType	0	1

### < 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.

Child Elements	Type	MinOccurs	MaxOccurs
show_subcommand	oval-def:EntityObjectStringType	1	1

### < 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 the sub-command and the corresponding config line. Please refer to the individual elements in the schema for more details about what each represents.

Child Elements	Type	MinOccurs	MaxOccurs
show_subcommand	oval-def:EntityStateStringType	0	1
config_line	oval-def:EntityStateStringType	0	1

---

---

### < snmp\_test >

Tests if lines under the global context associated with snmp that have a specific access list or community

name.

Child Elements	Type	MinOccurs	MaxOccurs
object	oval-def:ObjectRefType	1	1
state	oval-def:StateRefType	0	1

### < snmp\_object >

The snmp\_object element is used by a snmp test to define those objects to be 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.

### < snmp\_state >

Child Elements	Type	MinOccurs	MaxOccurs
access_list	oval-def:EntityStateStringType	0	1
community_name	oval-def:EntityStateStringType	0	1

---

---

### < 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. The evaluation of the test is guided by the check attribute that is inherited from the TestType.

Child Elements	Type	MinOccurs	MaxOccurs
object	oval-def:ObjectRefType	1	1
state	oval-def:StateRefType	0	1

### < tclsh\_object >

The tclsh\_object element is used by a tclsh test to define those objects to be evaluated based on a specified state. There is actually only one object relating to tclsh 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.

### < 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.

Child Elements	Type	MinOccurs	MaxOccurs
available	oval-def:EntityStateBoolType	0	1

---

---

### < version\_test >

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. The evaluation of the test is guided by the check attribute that is inherited from the TestType.

Child Elements	Type	MinOccurs	MaxOccurs
object	oval-def:ObjectRefType	1	1
state	oval-def:StateRefType	0	1

### < version\_object >

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.

### < version\_state >

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.

Child Elements	Type	MinOccurs	MaxOccurs
major_release	oval-def:EntityStateStringType	0	1
train_number	oval-def:EntityStateStringType	0	1
train_identifier	<a href="#">ios-def:EntityStateTrainIdentifierType</a>	0	1

version_string	oval-def:EntityStateStringType	0	1
----------------	--------------------------------	---	---

---



---

**== EntityStateTrainIdentifierType ==**

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.

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.