- Open Vulnerability and Assessment Language - Element Dictionary

- Schema: Windows 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 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 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.

< accesstoken_test >

The access token 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. 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

< accesstoken_object >

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 definied 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 access token object consists of a single security principle that identifies user, group, or compter account that is associated with the token.

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

< accesstoken_state >

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.

Child Elements	Type	MinOccurs	MaxOccurs
security_principle	oval-def:EntityStateStringType	0	1
seassignprimarytokenprivilege	oval-def:EntityStateBoolType	0	1
seauditprivilege	oval-def:EntityStateBoolType	0	1
sebackupprivilege	oval-def:EntityStateBoolType	0	1
sechangenotifyprivilege	oval-def:EntityStateBoolType	0	1
secreateglobalprivilege	oval-def:EntityStateBoolType	0	1
secreatepagefileprivilege	oval-def:EntityStateBoolType	0	1
secreatepermanentprivilege	oval-def:EntityStateBoolType	0	1
secreatetokenprivilege	oval-def:EntityStateBoolType	0	1
sedebugprivilege	oval-def:EntityStateBoolType	0	1
seenabledelegationprivilege	oval-def:EntityStateBoolType	0	1
seimpersonateprivilege	oval-def:EntityStateBoolType	0	1
seincreasebasepriorityprivilege	oval-def:EntityStateBoolType	0	1
seincreasequotaprivilege	oval-def:EntityStateBoolType	0	1
seloaddriverprivilege	oval-def:EntityStateBoolType	0	1
selockmemoryprivilege	oval-def:EntityStateBoolType	0	1
semachineaccountprivilege	oval-def:EntityStateBoolType	0	1
semanagevolumeprivilege	oval-def:EntityStateBoolType	0	1
seprofilesingleprocessprivilege	oval-def:EntityStateBoolType	0	1
seremoteshutdownprivilege	oval-def:EntityStateBoolType	0	1
serestoreprivilege	oval-def:EntityStateBoolType	0	1
sesecurityprivilege	oval-def:EntityStateBoolType	0	1
seshutdownprivilege	oval-def:EntityStateBoolType	0	1
sesyncagentprivilege	oval-def:EntityStateBoolType	0	1
sesystemenvironmentprivilege	oval-def:EntityStateBoolType	0	1

sesystemprofileprivilege	oval-def:EntityStateBoolType	0	1
sesystemtimeprivilege	oval-def:EntityStateBoolType	0	1
setakeownershipprivilege	oval-def:EntityStateBoolType	0	1
setcbprivilege	oval-def:EntityStateBoolType	0	1
seundockprivilege	oval-def:EntityStateBoolType	0	1
seunsolicitedinputprivilege	oval-def:EntityStateBoolType	0	1
sebatchlogonright	oval-def:EntityStateBoolType	0	1
seinteractivelogonright	oval-def:EntityStateBoolType	0	1
senetworklogonright	oval-def:EntityStateBoolType	0	1
seremoteinteractivelogonright	oval-def:EntityStateBoolType	0	1
seservicelogonright	oval-def:EntityStateBoolType	0	1
sedenybatchLogonright	oval-def:EntityStateBoolType	0	1
sedenyinteractivelogonright	oval-def:EntityStateBoolType	0	1
sedenynetworklogonright	oval-def:EntityStateBoolType	0	1
sedenyremoteInteractivelogonright	oval-def:EntityStateBoolType	0	1
sedenyservicelogonright	oval-def:EntityStateBoolType	0	1

< 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. 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

< 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 definied 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.

Child Elements	Type	MinOccurs	MaxOccurs
naming_context	win-def:EntityObjectNamingContextType	1	1
relative_dn	oval-def:EntityObjectStringType	1	1
attribute	oval-def:EntityObjectStringType	1	1

< 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.

Child Elements	Type	MinOccurs	MaxOccurs
naming_context	win-def:EntityStateNamingContextType	0	1
relative_dn	oval-def:EntityStateStringType	0	1
attribute	oval-def:EntityStateStringType	0	1
object_class	oval-def:EntityStateStringType	0	1
adstype	win-def:EntityStateAdstypeType	0	1
value	oval-def:EntityStateAnyType	0	1

< auditeventpolicy_test >

The audit event policy 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. 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

< 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.

< 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.

Child Elements	Type	MinOccurs	MaxOccurs
account_logon	win-def:EntityStateAuditType	0	1
account_management	win-def:EntityStateAuditType	0	1
detailed_tracking	win-def:EntityStateAuditType	0	1
directory_service_access	win-def:EntityStateAuditType	0	1
logon	win-def:EntityStateAuditType	0	1
object_access	win-def:EntityStateAuditType	0	1
policy_change	win-def:EntityStateAuditType	0	1
privilege_use	win-def:EntityStateAuditType	0	1
system	win-def:EntityStateAuditType	0	1

< 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. 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

< file_object >

The file_object element is used by a file test to define the specific file(s) to be evaluated. Each object extends the standard ObjectType as definied 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.

Child Elements	Type	MinOccurs	MaxOccurs
behaviors	win-def:FileBehaviors	0	1
path	oval-def:EntityObjectStringType	1	1
filename	oval-def:EntityObjectStringType	1	1

< 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.

Child Elements	Type	MinOccurs	MaxOccurs
path	oval-def:EntityStateStringType	0	1
filename	oval-def:EntityStateStringType	0	1
owner	oval-def:EntityStateStringType	0	1
size	oval-def:EntityStateIntType	0	1
a_time	oval-def:EntityStateStringType	0	1
c_time	oval-def:EntityStateStringType	0	1
m_time	oval-def:EntityStateStringType	0	1
ms_checksum	oval-def:EntityStateStringType	0	1
version	oval-def:EntityStateStringType	0	1
type	win-def:EntityStateFileTypeType	0	1
development_class	oval-def:EntityStateStringType	0	1

== FileBehaviors ==

The FileBehaviors complex type defines a number of behaviors that allow a more detailed definition of

the file objects being specified.

Attributes:

- max_depth n/a (optional -- default='-1')
- recurse_direction n/a (optional -- default='none')

< fileauditedpermissions_test >

The file audit permissions test is used to check the audit permissions 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 fileauditedpermissions_object and the optional state element specifies the metadata 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

< fileauditedpermissions_object >

The fileauditedpermissions_object element is used by a file audited permissions test to define the objects used to evalutate against the specified state. Each object extends the standard ObjectType as definied 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.

Child Elements	Type	MinOccurs	MaxOccurs
behaviors	win-def:FileAuditPermissionsBehaviors	0	1
path	oval-def:EntityObjectStringType	1	1
filename	oval-def:EntityObjectStringType	1	1
trustee_name	oval-def:EntityObjectStringType	1	1

< fileauditedpermissions_state >

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.

Child Elements	Type	MinOccurs	MaxOccurs
path	oval-def:EntityStateStringType	0	1
filename	oval-def:EntityStateStringType	0	1
trustee_name	oval-def:EntityStateStringType	0	1
standard_delete	win-def:EntityStateAuditType	0	1
standard_read_control	win-def:EntityStateAuditType	0	1
standard_write_dac	win-def:EntityStateAuditType	0	1
standard_write_owner	win-def:EntityStateAuditType	0	1
standard_synchronize	win-def:EntityStateAuditType	0	1
access_system_security	win-def:EntityStateAuditType	0	1
generic_read	win-def:EntityStateAuditType	0	1
generic_write	win-def:EntityStateAuditType	0	1
generic_execute	win-def:EntityStateAuditType	0	1
generic_all	win-def:EntityStateAuditType	0	1
file_read_data	win-def:EntityStateAuditType	0	1
file_write_data	win-def:EntityStateAuditType	0	1
file_append_data	win-def:EntityStateAuditType	0	1
file_read_ea	win-def:EntityStateAuditType	0	1
file_write_ea	win-def:EntityStateAuditType	0	1
file_execute	win-def:EntityStateAuditType	0	1
file_delete_child	win-def:EntityStateAuditType	0	1
file_read_attributes	win-def:EntityStateAuditType	0	1
file_write_attributes	win-def:EntityStateAuditType	0	1

== FileAuditPermissionsBehaviors ==

These behaviors allow a more detailed definition of the fileauditpermissions_objects being specified.

Attributes:

- max_depth n/a (optional -- default='-1')
- recurse_direction n/a (optional -- default='none')

< fileeffectiverights_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 ACL 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. 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

< fileeffectiverights_object >

The fileeffectiverights_object element is used by a file effective rights test to define the objects used to evalutate against the specified state. Each object extends the standard ObjectType as definied 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.

Child Elements	Type	MinOccurs	MaxOccurs
behaviors	win-def:FileEffectiveRightsBehaviors	0	1
path	oval-def:EntityObjectStringType	1	1
filename	oval-def:EntityObjectStringType	1	1
trustee_name	oval-def:EntityObjectStringType	1	1

< fileeffectiverights_state >

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.

Child Elements	Type	MinOccurs	MaxOccurs
path	oval-def:EntityStateStringType	0	1
filename	oval-def:EntityStateStringType	0	1
trustee_name	oval-def:EntityStateStringType	0	1
standard_delete	oval-def:EntityStateBoolType	0	1
standard_read_control	oval-def:EntityStateBoolType	0	1
standard_write_dac	oval-def:EntityStateBoolType	0	1
standard_write_owner	oval-def:EntityStateBoolType	0	1
standard_synchronize	oval-def:EntityStateBoolType	0	1
access_system_security	oval-def:EntityStateBoolType	0	1
generic_read	oval-def:EntityStateBoolType	0	1
generic_write	oval-def:EntityStateBoolType	0	1
generic_execute	oval-def:EntityStateBoolType	0	1
generic_all	oval-def:EntityStateBoolType	0	1
file_read_data	oval-def:EntityStateBoolType	0	1
file_write_data	oval-def:EntityStateBoolType	0	1
file_append_data	oval-def:EntityStateBoolType	0	1
file_read_ea	oval-def:EntityStateBoolType	0	1
file_write_ea	oval-def:EntityStateBoolType	0	1
file_execute	oval-def:EntityStateBoolType	0	1
file_delete_child	oval-def:EntityStateBoolType	0	1
file_read_attributes	oval-def:EntityStateBoolType	0	1
file_write_attributes	oval-def:EntityStateBoolType	0	1

== FileEffectiveRightsBehaviors ==

These behaviors allow a more detailed definition of the fileeffectiverights_objects being specified.

Attributes:

- max_depth n/a (optional -- default='-1')
- recurse_direction n/a (optional -- default='none')
- include_group xsd:boolean (optional -- default='true')
- resolve_group xsd:boolean (optional -- default='false')

< group_test >

The group test allows the different users that belong to specific groups 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 group_object and the optional state element specifies the metadata 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

< group_object >

The group_object element is used by a group test to define the specific group(s) to be evaluated. Each object extends the standard ObjectType as definied 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
group	oval-def:EntityObjectStringType	1	1

< group_state >

The group_state element enumerates the different users associate with a Windows group. Please refer to the individual elements in the schema for more details about what each represents.

Child Elements	Type	MinOccurs	MaxOccurs
group	oval-def:EntityStateStringType	0	1
user	oval-def:EntityStateStringType	0	1

< 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. 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

< 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 definied 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.

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

< 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.

Child Elements	Type	MinOccurs	MaxOccurs
name	oval-def:EntityStateStringType	0	1
index	oval-def:EntityStateIntType	0	1
type	win-def:EntityStateInterfaceTypeType	0	1
hardware_addr	oval-def:EntityStateStringType	0	1
inet_addr	oval-def:EntityStateStringType	0	1
broadcast_addr	oval-def:EntityStateStringType	0	1
netmask	oval-def:EntityStateStringType	0	1
addr_type	win-def:EntityStateAddrTypeType	0	1

< 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. 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

< 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.

< 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.

Child Elements	Type	MinOccurs	MaxOccurs
force_logoff	oval-def:EntityStateIntType	0	1
lockout_duration	oval-def:EntityStateIntType	0	1
lockout_observation_window	oval-def:EntityStateIntType	0	1
lockout_threshold	oval-def:EntityStateIntType	0	1

< 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. 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

< 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 definied 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).

Child Elements	Type	MinOccurs	MaxOccurs
key	oval-def:EntityObjectStringType	1	1
id	oval-def:EntityObjectIntType	1	1

< 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.

Child Elements	Type	MinOccurs	MaxOccurs
key	oval-def:EntityStateStringType	0	1
id	oval-def:EntityStateIntType	0	1
name	oval-def:EntityStateStringType	0	1
user_type	oval-def:EntityStateStringType	0	1
data_type	oval-def:EntityStateStringType	0	1
data	oval-def:EntityStateAnyType	0	1

< passwordpolicy_test >

The password policy test is used to check specific policy associated with passwords. 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. The evaluation of the test is guided by the check

attribute that is inherited from the TestType.

NOTE: This information is stored in the SAM or Active Directory but is encrypted or hidden so the registry_test and activedirectory_test are of no use. If this can be figured out, then the password_policy test is not needed.

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

< 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.

< 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.

Child Elements	Type	MinOccurs	MaxOccurs
max_passwd_age	oval-def:EntityStateIntType	0	1
min_passwd_age	oval-def:EntityStateIntType	0	1
min_passwd_len	oval-def:EntityStateIntType	0	1
password_hist_len	oval-def:EntityStateIntType	0	1
password_complexity	oval-def:EntityStateBoolType	0	1
reversible_encryption	oval-def:EntityStateBoolType	0	1

< 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. 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

< 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 definied 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).

Child Elements	Type	MinOccurs	MaxOccurs
local_address	oval-def:EntityObjectStringType	1	1
local_port	oval-def:EntityObjectIntType	1	1
protocol	win-def:EntityObjectProtocolType	1	1

< 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.

Child Elements	Type	MinOccurs	MaxOccurs
local_address	oval-def:EntityStateStringType	0	1
local_port	oval-def:EntityStateIntType	0	1
protocol	win-def:EntityStateProtocolType	0	1
pid	oval-def:EntityStateIntType	0	1

< process_test >

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 specifies the process information 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

< process_object >

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 definied 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(s).

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

< process_state >

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.

Child Elements	Type	MinOccurs	MaxOccurs
command_line	oval-def:EntityStateStringType	0	1
pid	oval-def:EntityStateIntType	0	1
ppid	oval-def:EntityStateIntType	0	1
priority	oval-def:EntityStateStringType	0	1
image_path	oval-def:EntityStateStringType	0	1
current_dir	oval-def:EntityStateStringType	0	1

< 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. 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

< registry_object >

Child Elements	Type	MinOccurs	MaxOccurs
behaviors	win-def:RegistryBehaviors	0	1
hive	win-def:EntityObjectRegistryHiveType	1	1
key	oval-def:EntityObjectStringType	1	1
name	oval-def:EntityObjectStringType	1	1

< 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.

Child Elements	Type	MinOccurs	MaxOccurs
hive	win-def:EntityStateRegistryHiveType	0	1
key	oval-def:EntityStateStringType	0	1
name	oval-def:EntityStateStringType	0	1
type	win-def:EntityStateRegistryTypeType	0	1
value	oval-def:EntityStateAnyType	0	1

== RegistryBehaviors ==

The RegistryBehaviors complex type defines a number of behaviors that allow a more detailed definition of the registry objects being specified.

Attributes:

- max_depth n/a (optional -- default='-1')
- recurse_direction n/a (optional -- default='none')

< regkeyauditedpermissions_test >

The registry key audited permissions test is used to check the audit permissions associated with Windows registry 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 regkeyauditedpermissions_object and the optional state element specifies the metadata 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

< regkeyauditedpermissions_object >

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 definied 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.

Child Elements	Type	MinOccurs	MaxOccurs
behaviors	win-def:RegkeyAuditPermissionsBehaviors	0	1
hive	win-def:EntityObjectRegistryHiveType	1	1
key	oval-def:EntityObjectStringType	1	1
trustee_name	oval-def:EntityObjectStringType	1	1

< regkeyauditedpermissions_state >

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.

Child Elements	Type	MinOccurs	MaxOccurs

hive	win-def:EntityStateRegistryHiveType	0	1
key	oval-def:EntityStateStringType	0	1
trustee_name	oval-def:EntityStateStringType	0	1
standard_delete	win-def:EntityStateAuditType	0	1
standard_read_control	win-def:EntityStateAuditType	0	1
standard_write_dac	win-def:EntityStateAuditType	0	1
standard_write_owner	win-def:EntityStateAuditType	0	1
standard_synchronize	win-def:EntityStateAuditType	0	1
access_system_security	win-def:EntityStateAuditType	0	1
generic_read	win-def:EntityStateAuditType	0	1
generic_write	win-def:EntityStateAuditType	0	1
generic_execute	win-def:EntityStateAuditType	0	1
generic_all	win-def:EntityStateAuditType	0	1
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
key_create_link	win-def:EntityStateAuditType	0	1
key_wow64_64key	win-def:EntityStateAuditType	0	1
key_wow64_32key	win-def:EntityStateAuditType	0	1
key_wow64_res	win-def:EntityStateAuditType	0	1

== RegkeyAuditPermissionsBehaviors ==

The RegkeyAuditPermissionsBehaviors complex type defines a number of behaviors that allow a more detailed definition of the registrykeyauditedpermissions objects being specified.

Attributes:

- max_depth n/a (optional -- default='-1')
- recurse_direction n/a (optional -- default='none')

< regkeyeffectiverights_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 ACL 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. 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

< regkeyeffectiverights_object >

Child Elements	Type	MinOccurs	MaxOccurs
behaviors	win-def:RegkeyEffectiveRightsBehaviors	0	1
hive	win-def:EntityObjectRegistryHiveType	1	1
key	oval-def:EntityObjectStringType	1	1
trustee_name	oval-def:EntityObjectStringType	1	1

< regkeyeffectiverights_state >

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.

Child Elements	Type	MinOccurs	MaxOccurs
hive	win-def:EntityStateRegistryHiveType	0	1
key	oval-def:EntityStateStringType	0	1
trustee_name	oval-def:EntityStateStringType	0	1
standard_delete	oval-def:EntityStateBoolType	0	1
standard_read_control	oval-def:EntityStateBoolType	0	1
standard_write_dac	oval-def:EntityStateBoolType	0	1
standard_write_owner	oval-def:EntityStateBoolType	0	1
standard_synchronize	oval-def:EntityStateBoolType	0	1
access_system_security	oval-def:EntityStateBoolType	0	1
generic_read	oval-def:EntityStateBoolType	0	1
generic_write	oval-def:EntityStateBoolType	0	1

generic_execute	oval-def:EntityStateBoolType	0	1
generic_all	oval-def:EntityStateBoolType	0	1
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	0	1
key_create_link	oval-def:EntityStateBoolType	0	1
key_wow64_64key	oval-def:EntityStateBoolType	0	1
key_wow64_32key	oval-def:EntityStateBoolType	0	1
key_wow64_res	oval-def:EntityStateBoolType	0	1

== RegkeyEffectiveRightsBehaviors ==

The RegkeyEffectiveRightsBehaviors complex type defines a number of behaviors that allow a more detailed definition of the registrykeyeffectiverights objects being specified.

Attributes:

- max_depth n/a (optional -- default='-1')
- recurse_direction n/a (optional -- default='none')

< 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. 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

< sid_object >

Child Elements	Type	MinOccurs	MaxOccurs
behaviors	win-def:SidBehaviors	0	1
trustee_name	oval-def:EntityObjectStringType	1	1

< sid_state >

Child Elements	Type	MinOccurs	MaxOccurs
trustee_name	oval-def:EntityStateStringType	0	1
trustee_sid	oval-def:EntityStateStringType	0	1
trustee_domain	oval-def:EntityStateStringType	0	1

== SidBehaviors ==

The SidBehaviors complex type defines a number of behaviors that allow a more detailed definition of the sid objects being specified.

Attributes:

- include_group xsd:boolean (optional -- default='true')
- resolve_group xsd:boolean (optional -- default='false')

< user_test >

The user test is used to check information fabout Windows users. 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. 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

< user_object >

Child Elements	Type	MinOccurs	MaxOccurs

user	oval-def:EntityObjectStringType	1	1

< user_state >

Child Elements	Type	MinOccurs	MaxOccurs
user	oval-def:EntityStateStringType	0	1
enabled	oval-def:EntityStateBoolType	0	1
group	oval-def:EntityStateStringType	0	1

< volume_test >

The volume test is used to check information about different storage volumes found 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 volume_object and the optional state element specifies the metadata 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

< 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 definied 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).

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

< volume_state >

The volume_state element defines the different metadata associate with a storage volume in Windows.

This includes the rootpat, 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.

Child Elements	Type	MinOccurs	MaxOccurs
rootpath	oval-def:EntityStateStringType	0	1
file_system	oval-def:EntityStateStringType	0	1
name	oval-def:EntityStateStringType	0	1
volume_max_component_length	oval-def:EntityStateIntType	0	1
serial_number	oval-def:EntityStateIntType	0	1
file_case_sensitive_search	oval-def:EntityStateBoolType	0	1
file_case_preserved_names	oval-def:EntityStateBoolType	0	1
file_unicode_on_disk	oval-def:EntityStateBoolType	0	1
file_persistent_acls	oval-def:EntityStateBoolType	0	1
file_file_compression	oval-def:EntityStateBoolType	0	1
file_volume_quotas	oval-def:EntityStateBoolType	0	1
file_supports_sparse_files	oval-def:EntityStateBoolType	0	1
file_supports_reparse_points	oval-def:EntityStateBoolType	0	1
file_supports_remote_storage	oval-def:EntityStateBoolType	0	1
file_volume_is_compressed	oval-def:EntityStateBoolType	0	1
file_supports_object_ids	oval-def:EntityStateBoolType	0	1
file_supports_encryption	oval-def:EntityStateBoolType	0	1
file_named_streams	oval-def:EntityStateBoolType	0	1
file_read_only_volume	oval-def:EntityStateBoolType	0	1

< wmi_test >

The wmi test is used to check information access 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. 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

< wmi_object >

Child Elements	Type	MinOccurs	MaxOccurs
namespace	oval-def:EntityObjectStringType	1	1
wql	oval-def:EntityObjectStringType	1	1

< wmi_state >

Child Elements	Type	MinOccurs	MaxOccurs
namespace	oval-def:EntityStateStringType	0	1
wql	oval-def:EntityStateStringType	0	1
result	oval-def:EntityStateAnyType	0	1

== 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 emlement associated with variable references.

Value	Description
MIB_IPADDR_DELETED	The address is being deleted.
MIB_IPADDR_DISCONNECTED	The address is on disconnected interface.
MIB_IPADDR_DYNAMIC	The stated address is a dynamic IP address.
MIB_IPADDR_PRIMARY	The stated address is a primary IP address.
MIB_IPADDR_TRANSIENT	The stated address is a transient address.

== 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 ADSTYPEENUM enumeration defined by Microsoft. The empty string is also allowed to support empty emlement associated with variable references.

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_STRIN	The string is of the case-insensitive type.
ADSTYPE_PRINTABLE_STRING	The string is displayable on screen or in print.
ADSTYPE_NUMERIC_STRING	The string is of a numeral 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.
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_DESCR	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.

== 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 emlement associated with variable references.

Value	Description
AUDIT_FAILURE	
AUDIT_NONE	
AUDIT_SUCCESS	
AUDIT_SUCCESS_FAILURE	

== 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 emlement associated with variable references.

Value	Description
MIB_IF_TYPE_ETHERNET	
MIB_IF_TYPE_FDDI	
MIB_IF_TYPE_LOOPBACK	
MIB_IF_TYPE_OTHER	
MIB_IF_TYPE_PPP	
MIB_IF_TYPE_SLIP	
MIB_IF_TYPE_TOKENRING	

== 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 emlement associated with variable references.

Value	Description
FILE_ATTRIBUTE_DIRECTORY	The handle identifies a directory.
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.

== 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 emlement associated with variable references.

Value	Description
domain	
configuration	
schema	

== EntityStateNamingContextType ==

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 emlement associated with variable references.

Value	Description
domain	
configuration	

schema

== 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 emlement associated with variable references.

Value	Description
TCP	
UDP	

== 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 emlement associated with variable references.

Value	Description
TCP	
UDP	

== EntityObjectRegistryHiveType ==

The EntityObjectRegistryHiveType 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 emlement associated with variable references.

Value	Description
HKEY_CLASSES_ROOT	
HKEY_CURRENT_CONFIG	
HKEY_CURRENT_USER	
HKEY_LOCAL_MACHINE	
HKEY_USERS	

== 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 emlement associated with variable references.

Value	Description
HKEY_CLASSES_ROOT	
HKEY_CURRENT_CONFIG	
HKEY_CURRENT_USER	
HKEY_LOCAL_MACHINE	
HKEY_USERS	

== EntityStateRegistryTypeType ==

The EntityStateRegistryTypeType complex type restricts a string value to a specific set of values. These values describe the possible types of data stored in a registry key. The empty string is also allowed to support empty emlement associated with variable references.

Value	Description
reg_binary	Binary data in any form.
reg_dword	A 32-bit number.
reg_expand_sz	Null-terminated string that contains unexpanded references to environment variables (for example, "%PATH%").
reg_multi_sz	Array of null-terminated strings, terminated by two null characters.
reg_qword	A 64-bit number.
reg_sz	Null-terminated string.