# - Open Vulnerability and Assessment Language - Element Dictionary

- Schema: Red Hat Definition
- Version: 4.2
- Release Date: 2 December 2005

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

---

## Elements

This section describes all the elements that are found within the schema, starting with the root element. Note that in the tables outlining possible attributes and child elements, square brackets [] means that the item is optional. All complex and simple types, along with attribute groups are described later in this document.

---

## File Test

---

### <file_test>

This test has been deprecated in version 4.1 of the redhat-schema and will be removed completely in version 5. It is recommended that all future OVAL Content use the file_test found in the unix-schema.

This test's purpose is to check a file's metadata, of the sort returned by either an ls command, stat command or stat() system call. The object being tested here is specified by an absolute path to a file. Remember that the default operator is EXISTS, so if no operator attribute is present for the path element and file specified is not found, then the test should fail.

| Extends: | standardTestType |
| --- | --- |
| | |

| | |
|---|---|
| Valid Sections: | notes, object, data |

---

# object section

---

### <path>

Specifies the absolute path to a file on the machine. This path can be created from multiple components that are added together. When a pattern match operator is used, the corresponding regular expression is matched against the set of absolute path strings. These string would not include the '.' and '..' notations. This means that a '.*' component of a regular expression will not only match all files in the specified directories, but all subdirectories, their subdirectories, etc.

| | |
|---|---|
| Parent Test: | File Test |
| Cardinality: | 1 |
| Content: | none |
| Valid Datatypes: | component |
| Valid Operators: | equals, not equal, pattern match |

# data section

---

### <type>

This is the file's type: regular file (regular), directory, named pipe (fifo), symbolic link, socket or block special.

| | |
|---|---|
| Parent Test: | File Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <group_id>

This is the group owner of the file, by group number.

| | |
|---|---|

| | |
|---|---|
| Parent Test: | File Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <user_id>

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.

| | |
|---|---|
| Parent Test: | File Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <a_time>

This is the time of the last access, in seconds since the last epoch.

| | |
|---|---|
| Parent Test: | File Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, greater than, less than, greater than or equal, less than or equal, pattern match |

### <c_time>

This is the time of the last change to the file's inode, which stores all.

| | |
|---|---|
| Parent Test: | File Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, greater than, less than, greater than or equal, less than or equal, pattern match |

### <m_time>

This is the time of the last change to the file's contents.

| | |
|---|---|
| Parent Test: | File Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, greater than, less than, greater than or equal, less than or equal, pattern match |

### <md5>

This is the MD5 hash of the file's contents, which serves as a kind of content integrity check.

| | |
|---|---|
| Parent Test: | File Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

---

# Inet Listening Servers Test

### <inetlisteningservers_test>

This test's purpose is generally used to check if a program is listening on the network, either for a new connections or as part of an ongoing connection. It is generally speaking the parsed output of running the command netstat -tuwlnpe with root privilege.

| | |
|---|---|
| Extends: | standardTestType |
| Valid Sections: | notes, object, data |

---

# object section

### <program_name>

This is the name of the communicating program.

| | |
|---|---|
| Parent Test: | Inet Listening Servers Test |
| Cardinality: | 1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

# data section

---

### <local_address>

This is the IP address of the network interface on which the program listens.

| | |
|---|---|
| Parent Test: | Inet Listening Servers Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <local_full_address>

This is the IP address and network port on which the program listens, equivalent to local_address:local_port.

| | |
|---|---|
| Parent Test: | Inet Listening Servers Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <local_port>

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.

| | |
|---|---|
| Parent Test: | Inet Listening Servers Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <foreign_address>

This is the IP address with which the program is communicating, or with which it will communicate, in the case of a listening server.

| | |
|---|---|
| Parent Test: | Inet Listening Servers Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <foreign_full_address>

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.

| | |
|---|---|
| Parent Test: | Inet Listening Servers Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <foreign_port>

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 a *.

| | |
|---|---|
| Parent Test: | Inet Listening Servers Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <pid>

This is the process ID of the process. The process in question is that of the program communicating on the network.

| | |
|---|---|
| Parent Test: | Inet Listening Servers Test |
| Cardinality: | 0-1 |
| Content: | integer |
| Valid Datatypes: | integer |
| Valid Operators: | equals, not equal, greater than, less than, greater than or equal, less than or equal |

### <protocol>

This is the transport-layer protocol, in lowercase: tcp or udp.

| | |
|---|---|
| Parent Test: | Inet Listening Servers Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <user_id>

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.

| | |
|---|---|
| Parent Test: | Inet Listening Servers Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

---

---

# Interface Test

### <interface_test>

This test has been deprecated in version 4.1 of the redhat-schema and will be removed completely in version 5. It is recommended that all future OVAL Content use the interface_test found in the unix-schema.

This test presents information one would expect to acquire by running ifconfig to display information about a particular network interface.

| | |
|---|---|
| Extends: | standardTestType |
| Valid Sections: | notes, object, data |

---

# object section

---

---

### <name>

This is the interface (eth0, eth1, etc.) name to check.

| | |
|---|---|
| Parent Test: | Interface Test |
| Cardinality: | 1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

# data section

---

---

### <hardware_addr>

This is the hardware or MAC address of the physical network card.

| | |
|---|---|
| Parent Test: | Interface Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <inet_addr>

This is the IP address of the interface.

| | |
|---|---|
| Parent Test: | Interface Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <broadcast_addr>

This is the broadcast IP address for this interface's network, like 192.168.255.255.

| | |
|---|---|
| Parent Test: | Interface Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <netmask>

This is the bitmask used to calculate the inteface'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.

| | |
|---|---|
| Parent Test: | Interface Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <flag>

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.

| | |
|---|---|
| Parent Test: | Interface Test |
| Cardinality: | 0-n |
| Content: | string |
| | |

| | |
|---|---|
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

# Permission Test

### <permission_test>

This test has been deprecated in version 4.1 of the redhat-schema and will be removed completely in version 5. It is recommended that all future OVAL Content use the permission_test found in the unix-schema.

This test checks the permission bits on a file, returning 1 or 0 based on the content of the named permission bit. The permission bits of a file are part of the octal "mode" of the file, a number that can be gathered via the stat command, stat() system call, or ls command. Each octal digit of the mode is a 3-bit number (0-7). In the first digit's bits are the Set-UID, Set-GID and Sticky bits. The remaining three digits are the user, group and other digits, corresponding to the user owner of the file, the group owner of the file, and then every other user on the system. Within these digits, the first bit is the read bit, the second bit is the write bit, and the third bit is the execute bit.

| | |
|---|---|
| Extends: | standardTestType |
| Valid Sections: | notes, object, data |

# object section

### <path>

Specifies the absolute path to a file on the machine. This path can be created from multiple components that are added together. When a pattern match operator is used, the corresponding regular expression is matched against the set of absolute path strings. These string would not include the '.' and '..' notations. This means that a '.*' component of a regular expression will not only match all files in the specified directories, but all subdirectories, their subdirectories, etc.

| | |
|---|---|
| Parent Test: | Permission Test |
| Cardinality: | 1 |
| Content: | none |
| | |

| | |
|---|---|
| Valid Datatypes: | component |
| Valid Operators: | equals, not equal, pattern match |

# data section

---

### <gexec>

Can the group owner of the file execute it or, if a directory, change into the directory?

| | |
|---|---|
| Parent Test: | Permission Test |
| Cardinality: | 0-1 |
| Content: | boolean |
| Valid Datatypes: | boolean |
| Valid Operators: | equals, not equal |

### <gread>

Can the group owner of the file read this file or, if a directory, read the directory contents?

| | |
|---|---|
| Parent Test: | Permission Test |
| Cardinality: | 0-1 |
| Content: | boolean |
| Valid Datatypes: | boolean |
| Valid Operators: | equals, not equal |

### <gwrite>

Can the group owner of the file write to this file or directory?

| | |
|---|---|
| Parent Test: | Permission Test |
| Cardinality: | 0-1 |
| Content: | boolean |
| Valid Datatypes: | boolean |
| Valid Operators: | equals, not equal |

### <oexec>

Can the other users execute this file or, if a directory, change into the directory?

| Parent Test: | Permission Test |
|---|---|
| Cardinality: | 0-1 |
| Content: | boolean |
| Valid Datatypes: | boolean |
| Valid Operators: | equals, not equal |

### <oread>

Can all other users read this file or, if a directory, read the directory contents?

| Parent Test: | Permission Test |
|---|---|
| Cardinality: | 0-1 |
| Content: | boolean |
| Valid Datatypes: | boolean |
| Valid Operators: | equals, not equal |

### <owrite>

Can the other users write to this file or directory?

| Parent Test: | Permission Test |
|---|---|
| Cardinality: | 0-1 |
| Content: | boolean |
| Valid Datatypes: | boolean |
| Valid Operators: | equals, not equal |

### <sgid>

Does the program run with the gid (thus privileges) of the file's group owner, rather than the calling user's group?

| Parent Test: | Permission Test |
|---|---|
| Cardinality: | 0-1 |
| Content: | boolean |
| Valid Datatypes: | boolean |
| Valid Operators: | equals, not equal |

### <sticky>

Can users delete each other's files in this directory, when said directory is writable by those users?

| Parent Test: | Permission Test |
|---|---|
| Cardinality: | 0-1 |
| Content: | boolean |
| Valid Datatypes: | boolean |
| Valid Operators: | equals, not equal |

### <suid>

Does the program run with the uid (thus privileges) of the file's owner, rather than the calling user?

| Parent Test: | Permission Test |
|---|---|
| Cardinality: | 0-1 |
| Content: | boolean |
| Valid Datatypes: | boolean |
| Valid Operators: | equals, not equal |

### <uexec>

Can the owner (user owner) of the file execute it or, if a directory, change into the directory?

| Parent Test: | Permission Test |
|---|---|
| Cardinality: | 0-1 |
| Content: | boolean |
| Valid Datatypes: | boolean |
| Valid Operators: | equals, not equal |

### <uread>

Can the owner (user owner) of the file read this file or, if a directory, read the directory contents?

| Parent Test: | Permission Test |
|---|---|
| Cardinality: | 0-1 |
| Content: | boolean |
| Valid Datatypes: | boolean |
| | |

| | |
|---|---|
| Valid Operators: | equals, not equal |

### <uwrite>

Can the owner (user owner) of the file read this file or, if a directory, read the directory contents?

| | |
|---|---|
| Parent Test: | Permission Test |
| Cardinality: | 0-1 |
| Content: | boolean |
| Valid Datatypes: | boolean |
| Valid Operators: | equals, not equal |

---

## Process Test

### <process_test>

This test has been deprecated in version 4.1 of the redhat-schema and will be removed completely in version 5. It is recommended that all future OVAL Content use the process_test found in the unix-schema.

This test checks the process information for a given process. It is equivalent to parsing the output of ps -ecf.

| | |
|---|---|
| Extends: | standardTestType |
| Valid Sections: | notes, object, data |

---

## object section

### <command>

This specifies the command/program name to check.

| | |
|---|---|
| Parent Test: | Process Test |
| Cardinality: | 1 |
| | |

| | |
|---|---|
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

# data section

---

### <exec_time>

This is the amount of CPU time (not clock time) that the process has consumed, formatted in HH:MM:SS or days.

| | |
|---|---|
| Parent Test: | Process Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <pid>

This is the process ID of the process.

| | |
|---|---|
| Parent Test: | Process Test |
| Cardinality: | 0-1 |
| Content: | integer |
| Valid Datatypes: | integer |
| Valid Operators: | equals, not equal, greater than, less than, greater than or equal, less than or equal |

### <ppid>

This is the process ID of the process's parent process.

| | |
|---|---|
| Parent Test: | Process Test |
| Cardinality: | 0-1 |
| Content: | integer |
| Valid Datatypes: | integer |
| | |

| | |
|---|---|
| Valid Operators: | equals, not equal, greater than, less than, greater than or equal, less than or equal |

### <priority>

This is the scheduling priority with which the process runs. This can be adjusted with the nice command or nice() system call.

| | |
|---|---|
| Parent Test: | Process Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <scheduling_class>

A characteristic maintained by the scheduler: RT (real-time), TS (timeshare), B (batch), BC (batch critical), WL (weightless) and GN (gang scheduled).

| | |
|---|---|
| Parent Test: | Process Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <start_time>

This is the time of day in which the process was started in either HH:MM:SS or days.

| | |
|---|---|
| Parent Test: | Process Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <tty>

This is the TTY on which the process was started, if applicable.

| | |
|---|---|
| Parent Test: | Process Test |

| | |
|---|---|
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

**<user_id>**

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.

| | |
|---|---|
| Parent Test: | Process Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

---

# RPM Info Test

**<rpminfo_test>**

This test checks the RPM header information for a given RPM package and should be fairly similar to calling rpm -qi package_name. Most applications actually use rpmversioncompare_test, as it is designed to compare the installed RPM's version information to an up-to-date or vulnearable RPM version.

| | |
|---|---|
| Extends: | standardTestType |
| Valid Sections: | notes, object, data |

---

# object section

**<name>**

This is the package name to check.

| | |
|---|---|

| | |
|---|---|
| Parent Test: | RPM Info Test |
| Cardinality: | 1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

# data section

---

### <arch>

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.

| | |
|---|---|
| Parent Test: | RPM Info Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <epoch>

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. 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

| | |
|---|---|
| Parent Test: | RPM Info Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <release>

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 21.11.4.

| | |
|---|---|

| | |
|---|---|
| Parent Test: | RPM Info Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <version>

This is the version number of the software built in this RPM. In the case of an apache rpm named httpd-2.0.40-21.11.4.i686.rpm, this value would be 2.0.40. This is the version number assigned by the apache code maintainers.

| | |
|---|---|
| Parent Test: | RPM Info Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <evr_version>

This represents the epoch, version, and release fields as a single version string. It has the form "EPOCH:VERSION-RELEASE".

| | |
|---|---|
| Parent Test: | RPM Info Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, greater than, less than, greater than or equal, less than or equal, pattern match |

### <signature_keyid>

This field is used to see if the RMP has been signed by the expected party.

| | |
|---|---|
| Parent Test: | RPM Info Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

# RPM Version Compare Test

### <rpmversioncompare_test>

This test has been deprecated in version 4.1 of the redhat-schema and will be removed completely in version 5. It is recommended that all future OVAL Content use the rpminfo_test.

This test checks the installed RPM against a given epoch, version and release number. This is the primary Red Hat schema item used to check the presence of unpatched software. To test for software before a given update, you simply plug in the information for the first RPM that doesn't have the given security flaw.

| | |
|---|---|
| Extends: | standardTestType |
| Valid Sections: | notes, object, data |

# object section

### <name>

This is the package name to check.

| | |
|---|---|
| Parent Test: | RPM Version Compare Test |
| Cardinality: | 1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <tested_epoch>

This is the epoch number to test against. The epoch is used as a kludge for version-release comparisons where the vendor has done some kind of re-numbering or version forking. 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

| | |
|---|---|
| Parent Test: | RPM Version Compare Test |
| Cardinality: | 1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <tested_version>

This is the version number of the software that we want to test against. In the case of an apache rpm named httpd-2.0.40-21.11.4.i686.rpm, this value would be 2.0.40. This is the version number assigned by the apache code maintainers.

| | |
|---|---|
| Parent Test: | RPM Version Compare Test |
| Cardinality: | 1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <tested_release>

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 21.11.4.

| | |
|---|---|
| Parent Test: | RPM Version Compare Test |
| Cardinality: | 1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

## data section

---

---

### <installed_version>

This is the result of the comparison: earlier, equal, later or not installed.

| | |
|---|---|
| Parent Test: | RPM Version Compare Test |
| Cardinality: | 0-1 |

| | |
|---|---|
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

# Shadow Test

### <shadow_test>

This test allows you 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. Background: 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 near running 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.

| | |
|---|---|
| Extends: | standardTestType |
| Valid Sections: | notes, object, data |

# object section

### <username>

This is the name of the user being checked.

| | |
|---|---|
| Parent Test: | Shadow Test |
| Cardinality: | 1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

# data section

### <password>

This is the encrypted version of the user's password.

| | |
|---|---|
| Parent Test: | Shadow Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <chg_lst>

This is the date of the last password change in days since 1/1/1970.

| | |
|---|---|
| Parent Test: | Shadow Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <chg_allow>

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.

| | |
|---|---|
| Parent Test: | Shadow Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <chg_req>

This describes how long a user can keep a password before the system forces her to change it.

| | |
|---|---|
| Parent Test: | Shadow Test |
| Cardinality: | 0-1 |
| | |

| | |
|---|---|
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <exp_warn>

This describes how long before password expiration the system begins warning the user. The system will warn the user at each login.

| | |
|---|---|
| Parent Test: | Shadow Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <exp_inact>

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.

| | |
|---|---|
| Parent Test: | Shadow Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <exp_date>

This speicifies when will the account's password expire, in days since 1/1/1970.

| | |
|---|---|
| Parent Test: | Shadow Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <flag>

This is a reserved field that the shadow file may use in the future.

| | |
|---|---|
| Parent Test: | Shadow Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

# Text File Content Test

### <textfilecontent_test>

This test has been deprecated in version 4.1 of the redhat-schema and will be removed completely in version 5. It is recommended that all future OVAL Content use the textfilecontent_test found in the independent-schema.

This test allows you to check a file's content, basically by serving as a flexible, regular-expression enabled 'grep'. grep checks for the existence of a line matching a given pattern in a file.

| | |
|---|---|
| Extends: | standardTestType |
| Valid Sections: | notes, object, data |

# object section

### <path>

Specifies the absolute path to a file on the machine. This path can be created from multiple components that are added together. When a pattern match operator is used, the corresponding regular expression is matched against the set of absolute path strings. These string would not include the '.' and '..' notations. This means that a '.*' component of a regular expression will not only match all files in the specified directories, but all subdirectories, their subdirectories, etc.

| | |
|---|---|
| Parent Test: | Text File Content Test |
| Cardinality: | 1 |
| Content: | none |
| Valid Datatypes: | component |

| | |
|---|---|
| Valid Operators: | equals, not equal, pattern match |

**<line>**

The line element represents a line in the file and is represented using a regular expression.

| | |
|---|---|
| Parent Test: | Text File Content Test |
| Cardinality: | 1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | pattern match |

# data section

---

---

**<subexpression>**

Each subexpression in the regular expression of the line element is then tested against the value specified in the subexpression element.

| | |
|---|---|
| Parent Test: | Text File Content Test |
| Cardinality: | 0-n |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

---

---

# Uname Test

---

**<uname_test>**

This test has been deprecated in version 4.1 of the redhat-schema and will be removed completely in version 5. It is recommended that all future OVAL Content use the uname_test found in the unix-schema.

This 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"

| | |
|---|---|
| Extends: | standardTestType |
| Valid Sections: | notes, data |

---

# data section

---

### <machine_class>

This is the machine hardware name, 5th field from uname -a.

| | |
|---|---|
| Parent Test: | Uname Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <node_name>

This is the host name, the 2nd field from uname -a.

| | |
|---|---|
| Parent Test: | Uname Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <os_name>

This is the operating system name, the 1st field from uname -a.

| | |
|---|---|
| Parent Test: | Uname Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <os_release>

This is the build version, 4th field from uname -a. For example, from a running Linux system: "#1 Wed Aug 25 13:34:40 UTC 2004"

| | |
|---|---|
| Parent Test: | Uname Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <os_version>

This is the operating system version, the 3rd field from uname -a.

| | |
|---|---|
| Parent Test: | Uname Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

### <processor_type>

This is the processor type, 6th field from uname -a.

| | |
|---|---|
| Parent Test: | Uname Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

---

---

# XML File Content Test

---

### <xmlfilecontent_test>

This test has been deprecated in version 4.1 of the redhat-schema and will be removed completely in version 5. It is recommended that all future OVAL Content use the xmlfilecontent_test found in the independent-schema.

This test allows you to check an element in an XML file.

| | |
|---|---|
| Extends: | standardTestType |
| Valid Sections: | notes, object, data |

---

# object section

---

### <path>

Specifies the absolute path to a file on the machine. This path can be created from multiple components that are added together. When a pattern match operator is used, the corresponding regular expression is matched against the set of absolute path strings. These string would not include the '.' and '..' notations. This means that a '.*' component of a regular expression will not only match all files in the specified directories, but all subdirectories, their subdirectories, etc.

| | |
|---|---|
| Parent Test: | XML File Content Test |
| Cardinality: | 1 |
| Content: | none |
| Valid Datatypes: | component |
| Valid Operators: | equals, not equal, pattern match |

### <xpath>

Specifies an Xpath expression describing the nodes to look at.

| | |
|---|---|
| Parent Test: | XML File Content Test |
| Cardinality: | 1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

# data section

---

### <value_of>

The value element checks the value of the nodes found. How this is used is entirely controlled by operator attributes.

| | |
|---|---|
| Parent Test: | XML File Content Test |
| Cardinality: | 0-1 |
| Content: | string |
| Valid Datatypes: | string |
| Valid Operators: | equals, not equal, pattern match |

---

---

### <platform>

The valid platforms for the RedHat Linux family.

-- Red Hat Linux 9
-- Red Hat Enterprise Linux 3
-- Red Hat Enterprise Linux 4
-- Red Hat Fedora Core 1
-- Red Hat Fedora Core 2
-- Red Hat Fedora Core 3
-- Red Hat Fedora Core 4
-- Red Hat Fedora Core 5

---

# Complex Types

This section describes any global complex types defined in the schema. These types can be instantiated by elements in this schema as well as elements in other schemas. Note that in the tables outlining possible attributes and child elements, square brackets [] means that the item is optional.

---

### -- componentType --

The componentType allows a value to be obtained by combining pieces from different sources. Each string defined by the different component elements is concatenated together to form the final string used. Each child component element has an attribute called type. The value of this attribute determines where to get the string used to build the file path. A type of literal means to use the value of the child component element as is, and to just concatenated it to the other strings. If a pattern match operator has been specified with a componentType, then the final string should be thought of as the pattern to test. As of Version 4 of the OVAL schema, pattern match can not be specified for the idividual components.

| | |
|---|---|
| Extends: | oval:subtestBaseType |

| | |
|---|---|
| Attributes: | (includes oval:subtestAttributes) |
| Content: | none |
| Child Elements: | component |