- Open Vulnerability and Assessment Language - Element Dictionary

- Schema: Mac OS Definition
- Version: 4.2
- Release Date: 2 December 2005

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

Account Info Test

<accountinfo_test>

User account information (username, uid, gid, etc.) See netinfo(5) for field information, niutil(1) for retrieving it. We may need/want to add in data elements for things like authentication_authority, generateduid, mcx_settings (restricted account settings).

Extends:	standardTestType
Valid Sections:	notes, object, data

---

object section

<username>

Specifies the user of the account to gather information from.

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

data section

<password>

Obfuscated (*****) or encrypted password for this user.

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

<uid>

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:	Account Info Test
Cardinality:	0-1
Content:	integer
Valid Datatypes:	integer
	equals, not equal, greater than, less than, greater than or

Valid Operators:	equal, less than or equal

<gid>

Group ID of this account.

Parent Test:	Account Info 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

<realname>

User's real name, aka gecos field of /etc/passwd.

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

<home_dir>

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

<login_shell>

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

File Test

<file_test>

This test has been deprecated in version 4.1 of the macos-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 the full path to 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:	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 or netinfo dump. 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

<size>

This is the size of the file in bytes.

Parent Test:	File 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

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 macos-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 (en0, en1, fw0, 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

Nvram Test

<nvram_test>

This test pulls data from the 'nvram -p' output.

Extends:	standardTestType
Valid Sections:	notes, object, data

---

object section

<nvram_var>

This specifies the nvram variable to check.

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

data section

<nvram_value>

This is the value of the associated nvram variable.

Parent Test:	Nvram Test
Cardinality:	0-1
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 macos-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 members of the file's group 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 members of the file's group 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 members of the file's group 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 macos-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 the ps command.

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

Pw Policy Test

<pwpolicy_test>

This test pulls data from the 'pwpolicy -getpolicy' output. The actual values get stored under /var/db/netinfo/local.nidb/ in a Store.# file. Is this test actually needed, or can the text file content test be used instead?

Extends:	standardTestType
Valid Sections:	notes, object, data

object section

<username>

Parent Test:	Pw Policy Test
Cardinality:	1
Content:	string
Valid Datatypes:	string
Valid Operators:	equals

<userpass>

Parent Test:	Pw Policy Test
Cardinality:	1

Content:	string
Valid Datatypes:	string
Valid Operators:	equals

<directory_node>

Parent Test:	Pw Policy Test
Cardinality:	1
Content:	string
Valid Datatypes:	string
Valid Operators:	equals

data section

<maxChars>

Maximum number of characters allowed in a password.

Parent Test:	Pw Policy 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

<maxFailedLoginAttempts>

Maximum number of failed logins before the account is locked.

Parent Test:	Pw Policy Test
Cardinality:	0-1
Content:	integer
Valid Datatypes:	integer
Valid Operators:	exists, none exists, equals, not equal, pattern match

<minChars>

Minimum number of characters allowed in a password.

Parent Test:	Pw Policy 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

<passwordCannotBeName>

Defines if the password is allowed to be the same as the username or not

Parent Test:	Pw Policy Test
Cardinality:	0-1
Content:	boolean
Valid Datatypes:	boolean
Valid Operators:	equals, not equals

<requiresAlpha>

Defines if the password must contain an alphabetical character or not

Parent Test:	Pw Policy Test
Cardinality:	0-1
Content:	boolean
Valid Datatypes:	boolean
Valid Operators:	equals, not equals

<requiresNumeric>

Defines if the password must contain an numeric character or not

Parent Test:	Pw Policy Test
Cardinality:	0-1
Content:	boolean
Valid Datatypes:	boolean
Valid Operators:	equals, not equals

Text File Content Test

<textfilecontent_test>

This test has been deprecated in version 4.1 of the macos-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 macos-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: "Darwin TestHost 7.7.0 Darwin Kernel Version 7.7.0: Sun Nov 7 16:06:51 PST 2004; root:xnu/xnu-517.9.5.obj~1/RELEASE_PPC Power Macintosh powerpc"

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 Mac OS X system: "Darwin Kernel Version 7.7.0: Sun Nov 7 16:06:51 PST 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 macos-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 Mac OS X family.

-- Apple Mac OS X 10.3

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