# A Quick Look at the Structure of an OVAL™ Definition

Version 5.2

# Introduction

At its core, the OVAL Language is a simple representation of truth values associated with system components. Examples would be "the version of a file is 3.5.1" or "the value of the registry key is 10" or "the user FRED is a member of the DEPARTMENT group". This document will explain how these simple truth values can be represented in an OVAL Definition. A complete description of the language will not be provided, but rather a quick introduction that should enable users to understand the structure of an OVAL Definition.

# OVAL Overview

Open Vulnerability and Assessment Language (OVAL™) is an international, information security, community standard to promote open and publicly available security content, and to standardize the transfer of this information across the entire spectrum of security tools and services. OVAL includes a language used to encode system details, and an assortment of content repositories held throughout the community. The language standardizes the three main steps of the assessment process: representing configuration information of systems for testing; analyzing the system for the presence of the specified machine state (vulnerability, configuration, patch state, etc.); and reporting the results of this assessment. The repositories are collections of publicly available and open content that utilize the language.

The OVAL community has developed three schemas written in Extensible Markup Language (XML) to serve as the framework and vocabulary of the OVAL Language. These schemas correspond to the three steps of the assessment process: an OVAL System Characteristics schema for representing system information, an OVAL Definition schema for expressing a specific machine state, and an OVAL Results schema for reporting the results of an assessment.

Content written in the OVAL Language is located in one of the many repositories found within the community. One such repository, known as the OVAL Repository, is hosted by The MITRE Corporation. It is the central meeting place for the OVAL Community to discuss, analyze, store, and disseminate OVAL Definitions. Each definition in the OVAL Repository determines whether a specified software vulnerability, configuration issue, program, or patch is present on a system.

The information security community contributes to the development of OVAL by participating in the creation of the OVAL Language on the OVAL Developers Forum and by writing definitions for the OVAL Repository through the OVAL Community Forum. An OVAL Board consisting of representatives from a broad spectrum of industry, academia, and government organizations from around the world oversees and approves the OVAL Language and monitors the posting of the definitions hosted on the OVAL Web site. This means that the OVAL, which is funded by US-CERT at the U.S. Department of Homeland Security for the benefit of the community, reflects the insights and combined expertise of the broadest possible collection of security and system administration professionals worldwide.

# The Hello World Example

The example one is taught when learning a programming language is always 'Hello World'. It is essentially a very simple program that compiles and when run, produces the text 'Hello World' on the screen. Let's look at a similar example for OVAL by writing a definition to test that the (hypothetical) Windows registry key 'HKEY_LOCAL_MACHINE\SOFTWARE\oval\example' has a value equal to 'Hello World'.

The first step in creating our definition will be to create the actual OVAL Test that will define the registry key and value we want to examine. Once the test has been created, a definition is written to wrap that test in metadata.

## *The Registry Test*

A test in the OVAL Language is used to check the value of specified attributes related to a given object. The OVAL Test structure simply combines a reference for a given object (in this case a registry key) and a reference to the value (also known as state) we want to check.

Each test is given an ID to identify it. The check attribute helps determine the relationship between the object and state. For our hello world example we set the check attribute to 'all'. This means that the test will be true if ALL of the registry keys on the system that match our object declaration satisfy the values found in the state. Of course in our example only one registry key should match our object and we will check that this registry key has a value equal to "Hello World".

```
<registry_test id="oval:org.mitre.oval:tst:1" check="all">

    <object object_ref="oval:org.mitre.oval:obj:1"/>
    <state state_ref="oval:org.mitre.oval:ste:1"/>

</registry_test>
```

Let's dive a little deeper into the concepts of objects and states.

**The Registry Object**
Every test in OVAL needs to have one or more objects to check. Objects are actual 'things' on the system being evaluated. For example: a file, a user, a process, a registry key, etc. These objects are the pieces of the system that we are testing.

To specify a registry key in the OVAL Language, one creates a 'registry_object'. This is simply an xml representation of the data that uniquely identifies it. In this case, it consists of a hive, key, and name. For our hello world example we would create the following object:

```
<registry_object id="oval:org.mitre.oval:obj:1">

    <hive>HKEY_LOCAL_MACHINE</hive>
    <key>SOFTWARE\oval</key>
    <name>example</name>

</registry_object>
```

Notice that our registry object has an id associated with it and this id was referenced by our registry test.

**The Registry State**
Once we have defined the object we want to look at, the next step is to express the state of that object for the test to be considered TRUE. For our hello world example we would create a registry_state that in effect says "check that the registry we identified has a value of Hello World".

```
<registry_state id="oval:org.mitre.oval:ste:1">

    <value>Hello World</value>

</registry_state>
```

Again, notice that our registry state has an id associated with it and this id was referenced by our registry test.

## *The Definition*

A definition is the heart and soul of the OVAL Language. It is a definition that applications reference during system evaluation. The purpose of a definition is to combine one or more tests using logical operators AND and OR. In addition, it wraps metadata around these tests to help detail to a user what is going on.

```
<definition id="oval:org.mitre.oval:def:1">

    <metadata>...</metadata>
    <criteria>...</criteria>

</definition>
```

Each definition has an id associated with it just like tests, objects, and states. This id follows the same urn format as all the other ids with 'oval:' followed by a unique string (ideally in reverse dns form), followed by the three letter code 'def', and ending with an integer.

**The Metadata**
Associated with each definition is a set of metadata that provides descriptive text about what is being checked and how the definition should be used. The only required pieces of metadata are a title and a description. Additional metadata can be added if needed.

```
<metadata>
    <title>Hello World Example</title>
    <description>
          This definition is used to introduce the OVAL Language to individuals interested
          in writing OVAL Content.
    </description>
</metadata>
```

**The Criteria**
It is the criteria of an OVAL Definition that outlines what is being tested. The criteria contains all the individual tests and joins them together with AND and OR operators. This allows the individual writing the definition to express statements like "test that a file exists AND a registry key is equal to 10". Of course our example only looks at a single registry key so the criteria will only contain a single test. The example below shows how the criteria for our Hello World definition would be written.

```
<criteria>
    <criterion test_ref="oval:org.mitre.oval:tst:1" comment="the value of the registry key equals Hello World"/>
</criteria>
```

Each criteria contains individual criterion statements that reference a single test. The actual test is written separately and is referenced by the id identified by the test_ref attribute. In this case we will be looking at test *oval:org.mitre.oval:tst:1*. The comment associated with the criterion helps explain what the test being referenced will do.

## *The Complete Definition File*

Now that we have all the pieces, we can combine them to form an OVAL Definition File.

```
<oval_definitions>

    <definitions>

          <definition id="oval:org.mitre.oval:def:1">

                <metadata>
                      <title>Hello World Example</title>
                      <description>
                            This definition is used to introduce the OVAL Language to individuals interested
                            in writing OVAL Content.
                      </description>
                </metadata>

                <criteria>
                      <criterion test_ref="oval:org.mitre.oval:tst:1" comment="the value of the registry key equals Hello World"/>
                </criteria>

          </definition>

    </definitions>

    <tests>

          <registry_test id="oval:org.mitre.oval:tst:1" check="all">
                <object object_ref="oval:org.mitre.oval:obj:1"/>
                <state state_ref="oval:org.mitre.oval:ste:1"/>
          </registry_test>

    </tests>

    <objects>

          <registry_object id="oval:org.mitre.oval:obj:1">
                <hive>HKEY_LOCAL_MACHINE</hive>
                <key>SOFTWARE\oval</key>
```

```
                <name>example</name>
            </registry_object>

    </objects>

    <states>

            <registry_state id="oval:org.mitre.oval:ste:1">
                <value>Hello World</value>
            </registry_state>

    </states>

</oval_definitions>
```

# Additional Information

Note that the above OVAL Definition File is not technically valid. It does not include some of the more advanced features, as those features do not factor into the structure or the relationship of the different components. For additional information please read the more complete "Writing an OVAL Definition" document located on the OVAL Web site at http://oval.mitre.org or send an email to oval@mitre.org.