OVAL Functions

March 24, 2011

Matt Hansbury

HS SEDI | MITRE

Homeland Security Systems Engineering and Development Institute

OVAL Functions

Reminder of how functions work:

- Functions act on components (literal, variable, or object)
- Used to manipulate data referenced by the components
- Current functions (arithmetic, begin, concat, end, escape_regex, split, substring, time_difference, regex_capture)

```
<local_variable id="oval:example:var:1" version="1" comment="…"
                datatype="string">
  <split delimiter="-">
    <literal_component>a-a-a</literal_component>
  </split>
</local_variable>
```

Evaluates to:

```
<value>a</value>
<value>a</value>
<value<a</value>
```

Proposed Function - count

A function that, given one or more components, will calculate the number of values referenced by the components

- Result is captured in a local_variable
- Any multiple value components would have each of the values counted as one value in the calculation

```
<local_variable id="oval:example:var:1" version="1" comment="…"
                datatype="int">
  <count>
    <literal_component>AA</literal_component>
    <literal_component>BB</literal_component>
    <literal_component>CC</literal_component>
  </count>
</local_variable>
```

Evaluates to: 3

Proposed Function – count (example 2)

A more complex example, using a multiple value variable as a component.

```
<local_variable id="oval:example:var:11" version="1" comment="…"
                datatype="int">
  <count>
    <literal_component>AA</literal_component>
    <literal_component>BB</literal_component>
    <variable_component var_ref="oval:example:var:44"/>
  </count>
</local_variable>
<constant_variable id="oval:example:var:44" version="1" comment="…"
                   datatype="string">
  <value>CC</value>
  <value>BB</value>
  <value>EE</value>
</constant_variable>
```

Evaluates to: 5

Proposed Function – count (full example)

```
<definitions>
  <definition id="oval:example:def:1" version="1" class="miscellaneous">
    <criteria><criterion comment="Test that the count function is supported."
                         test_ref="oval:example:tst:1"/></criteria>
  </definition>
</definitions>
<tests>
  <variable_test id="oval:example:tst:1" version="1" comment="..." check_existence="all_exist" check="all">
    <object object_ref="oval:example:obj:1"/>
    <state state_ref="oval:example:ste:1"/>
  </variable_test>
</tests>
<objects>
  <variable_object " id="oval:example:obj:1" version="1">
    <var_ref>oval:example:var:1</var_ref>
  </variable_object>
  <registry_object xmlns="http://oval.mitre.org/XMLSchema/oval-definitions-5#windows" id="oval:example:obj:2"
                   version="1">
    <hive>HKEY_LOCAL_MACHINE</hive>
    <key>SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters</key>
    <name>NullSessionShares</name>
  </registry_object>
</objects>
<states>
   <variable_state comment="The state for the evaluation of the count function" id="oval:example:ste:1"
                   version="1">
      <value datatype="int" operation="equals">3</value>
   </variable_state>
</states>
<variables>
  <local_variable id="oval:example:var:1" version="1" comment="Count variable" datatype="int">
    <count>
      <object_component object_ref="oval:example:obj:2" item_field="value"/>
    </count>
  </local_variable>
</variables>
```

Proposed Function - unique

A function that, given one or more components, will return only those values that are unique.

- Essentially removes duplicates
- Result is captured in a local_variable
- Any multiple value components would have each of the values used as a value when computing duplicates

```
<local_variable id="oval:example:var:1" version="1" comment="…"
                datatype="string">
  <unique>
    <literal_component>AA</literal_component>
    <literal_component>BB</literal_component>
    <literal_component>BB</literal_component>
  </unique>
</local_variable>
```

Evaluates to:

```
<value>AA</value>
<value>BB</value>
```

Proposed Function – unique (example 2)

A more complex example, using a multiple value variable as a component.

```
<local_variable id="oval:example:var:22" version="1" comment="…"
                datatype="string">
  <unique>
    <literal_component>AA</literal_component>
    <literal_component>BB</literal_component>
    <literal_component>BB</literal_component>
    <variable_component var_ref="oval:example:var:55"/>
  </unique>
</local_variable>
<constant_variable id="oval:example:var:55" version="1"
                   comment="…datatype="string">
  <value>CC</value>
  <value>BB</value>
  <value>AA</value>
</constant_variable>
```

Evaluates to:

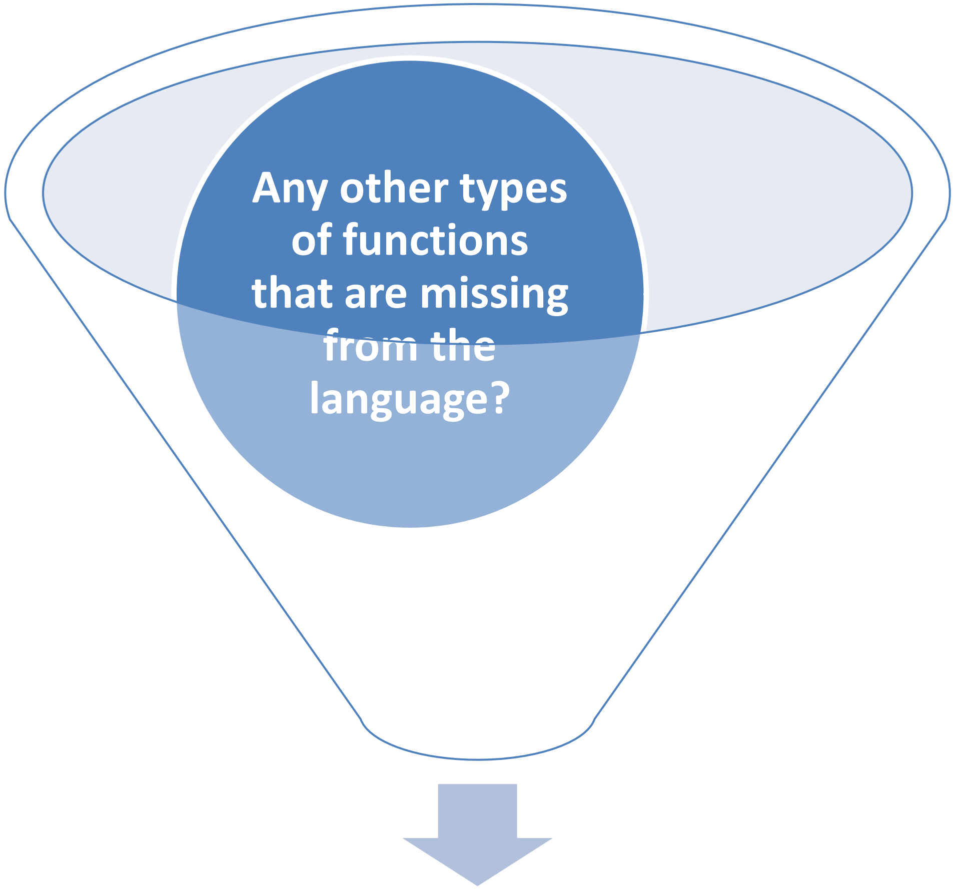
```
<value>AA</value>
<value>BB</value>
<value>CC</value>
```

Proposed Function – unique (full example)

```
<definitions>
  <definition id="oval:example:def:1" version="1" class="miscellaneous">
    <criteria><criterion comment="Test that the count function is supported."
                         test_ref="oval:example:tst:1"/></criteria>
  </definition>
</definitions>
<tests>
  <variable_test id="oval:example:tst:2" version="1" comment="..." check_existence="all_exist" check="all">
    <object object_ref="oval:example:obj:2"/>
    <state state_ref="oval:example
  </variable_test>
</tests>
<objects>
  <variable_object " id="oval:examp
     <var_ref>oval:example:var:1</var_ref>
  </variable_object>
  <registry_object xmlns="http://oval.mitre.org/XMLSchema/oval-definitions-5#windows" id="oval:example:obj:2"
                   version="1">
           …
  </registry_object>
</objects>
<states>
  <variable_state comment="The state for the evaluation of the count function" id="oval:example:ste:2"
                  version="1">
    <value datatype="string" operation="equals" var_check="all" var_ref="oval:example:var:3"/>
  </variable_state>
</states>
<variables>
   <local_variable id="oval:example:var:1" version="1" datatype="string" comment="Registry value variable">
    <unique>
      <object_component object_ref="oval:example:obj:2" item_field="value"/>
    </unique>
   </local_variable>
   <constant_variable id="oval:example:var:2" version="1" comment="Unique result variable" datatype="string">
    <value>AA</value>
    <value>BB</value>
   </constant_variable>
</variables>
```

```
<hive>HKEY_LOCAL_MACHINE</hive>
<key>SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters</key>
<name>NullSessionShares</name>
```

Other Functions?



Any other types of functions that are missing from the language?

HS SEDI | MITRE

Homeland Security Systems Engineering and Development Institute

Questions?

Summary

- Functions allow manipulation of data
- Two new proposed functions:
  - count – provides a count of all values specified by the components
  - unique – strips out duplicate values from a specified set of components
  - Consider other functions?

HS SEDI | MITRE
Homeland Security Systems Engineering and Development Institute