

# Service Logic Interpreter Nodes

## Table Of Contents

- [Table Of Contents](#)
- [Schema](#)
- [Supported node types](#)
  - [Flow Control](#)
    - [Block node](#)
    - [Break node](#)
    - [Call node](#)
    - [Exit node](#)
    - [For node](#)
    - [Return node](#)
    - [Set node](#)
    - [Switch node](#)
    - [While node](#)
  - [Device Management](#)
    - [Configure node](#)
  - [Java Plugin Support](#)
    - [Execute node](#)
  - [Recording](#)
    - [Record node](#)
  - [Resource Management](#)
    - [Delete node](#)
    - [Exists node](#)
    - [Get-resource node](#)
    - [Is-available node](#)
    - [Notify node](#)
    - [Release node](#)
    - [Reserve node](#)
    - [Save node](#)
    - [Update node](#)

## Schema

You can think of the JSON as Directed Graph source code and the XML as compiled code. The XML must conform to the schema at <https://gerrit.onap.org/r/gitweb?p=ccsdl/sli/core.git;a=blob;f=sli/common/src/main/resources/svclogic.xsd;hb=refs/heads/master>. The XML schema reveals what nodes are supported and which attributes are appropriate for each node. This page explains each node in plain English with accompanying code snippets as an alternative to reading the schema file itself.

## Supported node types

The following built-in node types are currently supported:

### Flow Control

#### Block node

##### Description

A **block** node is used to executes a set of nodes.

##### Attributes

<b>atomic</b>	if <i>true</i> , then if a node returns failure, subsequent nodes will not be executed and nodes already executed will be backed out. If unset it is treated as <i>false</i> .
---------------	--

##### Parameters

None

##### Outcomes

None

Example

```
<block>
  <record plugin="org.openecomp.sdnc.sli.recording.FileRecorder">
    <parameter name="file" value="/tmp/gamma_r1.log" />
    <parameter name="field1" value="__TIMESTAMP__" />
    <parameter name="field2" value="RESERVED"/>
    <parameter name="field3" value="$asePort.uni_circuit_id"/>
  </record>
  <return status="success">
    <parameter name="uni-circuit-id" value="$asePort.uni_circuit_id" />
  </return>
</block>
```

Break node

Description

A **break** node is used to break out of a for or while loop.

Attributes

None

Parameters

None

Outcomes

None

Example

```
<break/>
```

Call node

Description

A **call** node is used to call another graph

Attributes

<b>module</b>	Module of directed graph to call. If unset, defaults to that of calling graph
<b>rpc</b>	rpc of directed graph to call.
<b>version</b>	version of graph to call, If unset, uses active version.
<b>mode</b>	mode (sync/async) of graph to call. If unset, defaults to that of calling graph.

Parameters

Not applicable

Outcomes

<b>success</b>	Sub graph returned success
<b>not-found</b>	Graph not found
<b>failure</b>	Subgraph returned success

Example

```
<call rpc="svc-topology-reserve" mode="sync" />
```

## Exit node

### Description

An **exit** node is used to exit a graph without further execution

### Attributes

None

### Parameters

Not applicable

### Outcomes

None

### Example

```
<exit/>
```

## For node

### Description

A **for** node provides a fixed iteration looping mechanism, similar to the Java for loop

### Attributes

<b>atomic</b>	if <i>true</i> , then if a node returns failure, subsequent nodes will not be executed and nodes already executed will be backed out. If unset it is treated as <i>true</i> .
<b>index</b>	index variable
<b>start</b>	initial value
<b>end</b>	maximum value
<b>silentFailure</b>	Optional boolean, when true for loops will not throw an exception for a null or invalid index. If unset it is treated as <i>false</i> .


### Parameters

Not applicable.

### Outcomes

Not applicable. The **status** node has no outcomes.

### Example

```
 <for index="i" start="0" end="$network.num-segments">  
  <set>  
    <parameter name="$vlanlist" value="eval($vlanlist+', '+$network.segment[i].provider-segmentation-id)"/>  
  </set>  
</for>
```

## Return node

### Description

A **return** node is used to return a status to the invoking MD-SAL application. The return node does not prevent the execution of subsequent nodes. The status set by the return node is tested by atomic for and block nodes.

#### Attributes

<b>status</b>	Status value to return ( <i>success</i> or <i>failure</i> )
---------------	---

#### Parameters


The following optional parameters may be passed to convey more detailed status information.

<b>error-code</b>	A brief, usually numeric, code indicating the error condition
<b>error-message</b>	A more detailed error message

#### Outcomes

Not applicable. The **status** node has no outcomes.

#### Example

 `<return status="failure">  
 <parameter name="error-code" value="1542" />  
 <parameter name="error-message" value="Activation failure" />  
</return>`

## Set node

#### Description

A **set** node is used to set one or more values in the execution context

#### Attributes

<b>only-if-unset</b>	If true the set node will only execute if the current value of the target is null
----------------------	---

#### Parameters

Values to be set are passed as parameters

#### Outcomes

Not applicable. The **set** node has no outcomes.

#### Example

```
<set>  
  <parameter name="vlan" value="$network.provider-segmentation-id" />  
</set>
```

## Switch node

#### Description

A **switch** node is used to make a decision based on its **test** attribute.

#### Attributes

<b>test</b>	Condition to test
-------------	-------------------

#### Parameters

None

## Outcomes

Depends on the **test** condition

### Example

```
<switch test="$uni-cir-units">
  <outcome value="Mbps">
    <reserve plugin="org.openecomp.sdnc.sli.resource.gamma.GammaResource"
              resource="ase-port"
              key="resource-emt-clli == $edge-device-clli and speed >= $uni-cir-value"
              pfx="asePort">

    <outcome value="success">
      <return status="success">
        <parameter name="uni-circuit-id" value="$asePort.uni_circuit_id" />
      </return>
    </outcome>
    <outcome value="Other">
      <return status="failure">
        <parameter name="error-code" value="1010" />
        <parameter name="error-message" value="No ports found that match criteria" />
      </return>
    </outcome>
  </reserve>
</outcome>
<outcome value="Gbps">
  <reserve plugin="org.openecomp.sdnc.sli.resource.gamma.GammaResource"
            resource="ase-port"
            key="resource-emt-clli == $edge-device-clli and speed >= $uni-cir-value*1000"
            pfx="asePort">

  <outcome value="success">
    <return status="success">
      <parameter name="uni-circuit-id" value="$asePort.uni_circuit_id" />
    </return>
  </outcome>
  <outcome value="Other">
    <return status="failure">
      <parameter name="error-code" value="1010" />
      <parameter name="error-message" value="No ports found that match criteria" />
    </return>
  </outcome>
</reserve>
</outcome>
</switch>
```

## While node

### Description

A **while** node is used to create a while loop

### Attributes

<b>test</b>	Condition to test
<b>do</b>	Optional boolean, when true the loop executes once regardless of the test condition

### Parameters

None

## Outcomes

Depends on the **test** condition

### Example

## Device Management

### Configure node

#### Description

A **configure** node is used to configure a device.

#### Attributes

<b>adaptor</b>	Fully qualified Java class of resource adaptor to be used
<b>activate</b>	Activate device/interface, for devices that support a separate activation step.
<b>key</b>	SQL-like string specifying criteria for item to configure

#### Parameters

Specific to device adaptor.

#### Outcomes

<b>success</b>	Device successfully configured
<b>not-found</b>	Element to be configured does not exist.
<b>not-ready</b>	Element is not in a state where it can be configured/activated
<b>already-active</b>	Attempt to activate element that is already active
<b>failure</b>	Configure failed for some other reason

#### Example

```

<configure adaptor="org.openecomp.sdnc.sli.adaptor.emt.EmtAdaptor"
    key="$uni-circuit-id" activate="true">
  <parameter name="circuit.id" value="$uni-circuit-id" />
  <parameter name="subscriber.name" value="$subscriber-name" />
  <parameter name="emt.clii" value="$edge-device-clli" />
  <parameter name="port.tagging" value="$port-tagging" />
  <parameter name="port.mediaSpeed" value="$media-speed" />
  <parameter name="location.state" value="$uni-location-state" />
  <parameter name="location.city" value="$uni-location-city" />
  <parameter name="cosCategory" value="$cos-category" />
  <parameter name="gosProfile" value="$gos-profile" />
  <parameter name="lldp" value="$asePort.resource-lldp" />
  <parameter name="mtu" value="$asePort.resource-mtu" />
  <outcome value="success">
    <block>
      <record plugin="org.openecomp.sdnc.sli.recording.FileRecorder">
        <parameter name="file" value="/tmp/gamma_r1.log" />
        <parameter name="field1" value="__TIMESTAMP__" />
        <parameter name="field2" value="ACTIVE" />
        <parameter name="field3" value="$uni-circuit-id"/>
      </record>
      <return status="success">
        <parameter name="edge-device-clli" value="$asePort.resource-emt-clli" />
      </return>
    </block>
  </outcome>
  <outcome value="already-active">
    <return status="failure">
      <parameter name="error-code" value="1590" />
      <parameter name="error-message" value="Port already active" />
    </return>
  </outcome>
  <outcome value="Other">
    <return status="failure">
      <parameter name="error-code" value="1542" />
      <parameter name="error-message" value="Activation failure" />
    </return>
  </outcome>
</configure>

```

## Java Plugin Support

### Execute node

#### Description

An **execute** node is used to execute Java code supplied as a plugin

#### Attributes

\*\*\* Parameters

<b>plugin</b>	Fully qualified Java class of plugin to be used
<b>method</b>	Name of method in the plugin class to execute. Method must return void, and take 2 arguments: a Map (for parameters) and a SvcLogicContext (to allow plugin read/write access to context memory)
<b>emitsOutcome</b>	Optional boolean value, when true the plugin must return a String which should represent an outcome

Specific to plugin / method

#### Outcomes

<b>success</b>	Device successfully configured
<b>not-found</b>	Plugin class could not be loaded

<b>unsupported-method</b>	Named method taking (Map, SvcLogicContext) could not be found
<b>failure</b>	Configure failed for some other reason

### Example

```
<execute plugin="org.openecomp.sdnc.sli.plugin.HelloWorld"
  method="log">
  <parameter name="message" value="Hello, world!" />
  <outcome value="success">
    <return status="success" />
  </outcome>
  <outcome value="not-found">
    <return status="failure">
      <parameter name="error-code" value="1590" />
      <parameter name="error-message" value="Could not locate plugin" />
    </return>
  </outcome>
  <outcome value="Other">
    <return status="failure">
      <parameter name="error-code" value="1542" />
      <parameter name="error-message" value="Internal error" />
    </return>
  </outcome>
</execute>
```

## Recording

### Record node

#### Description

A **record** node is used to record an event. For example, this might be used to log provisioning events.

#### Attributes

<b>plugin</b>	Fully qualified Java class to handle recording.
---------------	---

#### Parameters

Parameters will depend on the plugin being used. For the FileRecorder class, the parameters are as follows

<b>file</b>	The file to which the record should be written
<b>field1</b>	First field to write. There will be <b>field</b> parameters for each field to write, from <b>field1</b> through <b>fieldN</b> . A special value <code>__TIMESTAMP__</code> may be assigned to a field to insert the current timestamp

#### Outcomes

<b>success</b>	Record successfully written
<b>failure</b>	Record could not be successfully written

### Example

```
<record plugin="org.openecomp.sdnc.sli.recording.FileRecorder">
  <parameter name="file" value="/tmp/gamma_r1.log" />
  <parameter name="field1" value="__TIMESTAMP__" />
  <parameter name="field2" value="ACTIVE" />
  <parameter name="field3" value="$uni-circuit-id" />
</record>
```

## Resource Management



## Delete node

### Description

A **delete** node is used to delete a resource from the local resource inventory.

### Attributes

<b>plugin</b>	Fully qualified Java class of resource adaptor to be used
<b>resource</b>	Type of resource to delete
<b>key</b>	SQL-like string specifying key to delete

### Parameters

None

### Outcomes

<b>success</b>	Resource specified deleted successfully.
<i>failure</i>	Resource specified was not deleted

### Example

```
<delete plugin="org.openecomp.sdnc.sli.resource.gamma.GammaResource"
        resource="ase-port"
        key="uni_circuit_id == $uni-circuit-id">
  <outcome value="true">
    <return status="success" />
  </outcome>
  <outcome value="false">
    <return status="failure" />
  </outcome>
</delete>
```

## Exists node

### Description

An **exists** node is used to determine whether a particular instance of a resource exists. For example, this might be used to test whether a particular switch CLLI is provisioned.

### Attributes

<b>plugin</b>	Fully qualified Java class of resource adaptor to be used
<b>resource</b>	Type of resource to check
<b>key</b>	SQL-like string specifying key to check for

### Parameters

None

### Outcomes

<b>true</b>	Resource specified exists.
<b>false</b>	Resource specified is unknown

### Example

```

<exists plugin="org.openecomp.sdnc.sli.resource.gamma.GammaResource"
  resource="ase-port"
  key="uni_circuit_id == $uni-circuit-id">
  <outcome value="true">
    <return status="success" />
  </outcome>
  <outcome value="false">
    <return status="failure" />
  </outcome>
</exists>

```

## Get-resource node

### Description

A **get-resource** node is used to retrieve information about a particular resource and make it available to other nodes in the service logic tree. For example, this might be used to retrieve information about a particular uni-port.

### Attributes

<b>plugin</b>	Fully qualified Java class of resource adaptor to be used
<b>resource</b>	Type of resource to retrieve
<b>key</b>	SQL-like string specifying criteria for retrieval
<b>pfx</b>	Prefix to add to context variable names set for data retrieved
<b>select</b>	String to specify, if key matches multiple entries, which entry should take precedence
<b>order-by</b>	Prefix to add to context variable names set for data retrieved

### Parameters

None

### Outcomes

<b>success</b>	Resource successfully retrieved
<b>not-found</b>	Resource referenced does not exist
<b>failure</b>	Resource retrieve failed for some other reason

### Example

```

<get-resource plugin="org.openecomp.sdnc.sli.resource.gamma.GammaResource"
  resource="ase-port"
  key="uni_circuit_id == $uni-circuit-id"
  pfx="current-port">
  <outcome value="success">
    <return status="success" />
  </outcome>
  <outcome value="not-found">
    <return status="failure" />
  </outcome>
  <outcome value="failure">
    <return status="failure" />
  </outcome>
</get-resource>

```

## Is-available node

### Description

An **is-available** node is used to determine whether a particular type of resource is available. For example, this might be used to test whether any ports are available for assignment on a particular switch.

Attributes

plugin	Fully qualified Java class of resource adaptor to be used
resource	Type of resource to check
key	SQL-like string specifying key to check for
pfx	Prefix to add to context variable names set for data retrieved

Parameters

None

Outcomes

true	Resource requested is available
false	Resource requested is not available

Example

```
<is-available plugin="org.openecomp.sdnc.sli.resource.gamma.GammaResource"
              resource="ase-port"
              key="resource-emt-clli == $edge-device-clli and speed >= $uni-cir-value">
  <outcome value="true">
    <return status="success" />
  </outcome>
  <outcome value="false">
    <return status="failure" />
  </outcome>
</is-available>
```

Notify node

Description

A **notify** node is used to inform an external application (e.g. A&AI) that a resource was updated.

Attributes

plugin	Fully qualified Java class of resource adaptor to be used
resource	Identifies resource that was updated
action	Action that triggered notification to be sent (ADD/UPDATE/DELETE)

Parameters

None

Outcomes

success	Notification was successful
failure	Notification failed is not available

Example

```
<notify plugin="org.openecomp.sdnc.sli.resource.gamma.GammaResource"
        resource="ase-port"
        action="ADD">
  <outcome value="success">
    <return status="success" />
  </outcome>
  <outcome value="Other">
    <return status="failure" />
  </outcome>
</notify>
```

## Release node

### Description

A **release** node is used to mark a resource as no longer in use, and thus available for assignment.

### Attributes

<b>plugin</b>	Fully qualified Java class of resource adaptor to be used
<b>resource</b>	Type of resource to release
<b>key</b>	SQL-like string specifying key to check of resource to release

### Parameters

None

### Outcomes

<b>success</b>	Resource successfully released
<b>not-found</b>	Resource referenced does not exist
<b>failure</b>	Resource release failed for some other reason

### Example

```
<release plugin="org.openecomp.sdnc.sli.resource.gamma.GammaResource"
        resource="ase-port"
        key="uni_circuit_id == $uni-circuit-id">
  <outcome value="success">
    <return status="success" />
  </outcome>
  <outcome value="not-found">
    <return status="failure" />
  </outcome>
  <outcome value="failure">
    <return status="failure" />
  </outcome>
</release>
```

## Reserve node

### Description

A **reserve** node is used to reserve a particular type of resource.. For example, this might be used to reserve a port on a particular switch.

### Attributes

<b>plugin</b>	Fully qualified Java class of resource adaptor to be used
<b>resource</b>	Type of resource to reserve

<b>key</b>	SQL-like string specifying criteria for reservation
<b>select</b>	String to specify, if <b>key</b> matches multiple entries, which entry should take precedence

## Parameters

None

## Outcomes

<b>success</b>	Resource requested was successfully reserved
<b>failure</b>	Resource requested was not successfully reserved

## Example

```
<reserve plugin="org.openecomp.sdnc.sli.resource.gamma.GammaResource"
  resource="ase-port"
  key="resource-emt-ctrl == $edge-device-ctrl and speed >= $uni-cir-value"
  select="min(speed)">
  <outcome value="success">
    <return status="success"/>
  </outcome>
  <outcome value="failure">
    <return status="failure"/>
  </outcome>
</reserve>
```

# Save node

## Description

A **save** node is used to save information about a particular resource to persistent storage. For example, this might be used to save information about a particular uni-port.

## Attributes

<b>plugin</b>	Fully qualified Java class of resource adaptor to be used
<b>resource</b>	Type of resource to save
<b>key</b>	SQL-like string specifying criteria for retrieval
<b>force</b>	If "true", save resource even if this resource is already stored in persistent storage
<b>pfx</b>	Prefix to be prepended to variable names, when attributes are set in SvcLogicContext

## Parameters

Values to save (columns) are specified as parameters, with each name corresponding to a column name and each value corresponding to the value to set.

## Outcomes

<b>success</b>	Resource successfully saved
<b>failure</b>	Resource save failed

## Example

```

<save plugin="\$gamma-resource-plugin\" resource="vnf"
    key="vnf-name = $requests.vnf.vnf-name" force="true"
    pfx="requests.vnf">
    <parameter name="vnf-name"
        value="\$requests.cust-country-code + $requests.cust-id + $requests.cust-city + $requests.cust-
state + '001VCE'" />
    <parameter name="vnf-type" value="vce" />
    <parameter name="orchestration-status" value="pending-create" />
    <parameter name="heat-stack-id" value="\$requests.heat-stack-id" />
    <parameter name="mso-catalog-key" value="\$requests.mso-catalog-key" />
    <parameter name="availability-zone"
        value="\$aic-availability-zone.availability-zone" />
    <parameter name="aic-site-id" value="\$requests.vnf.aic-site-id" />
    <parameter name="oam-ipv4-address" value="\$vce-ipv4-oam-addr.ipv4-addr" />
</save>

```

## Update node

### Description

An **update** node is used to update information about a particular resource to persistent storage.

### Attributes

<b>plugin</b>	Fully qualified Java class of resource adaptor to be used
<b>resource</b>	Type of resource to update
<b>key</b>	SQL-like string specifying criteria for retrieval
<b>pfx</b>	Prefix to be prepended to variable names, when attributes are set in SvcLogicContext

### Parameters

Values to save (columns) are specified as parameters, with each name corresponding to a column name and each value corresponding to the value to set.

### Outcomes

<b>success</b>	Resource successfully saved
<b>failure</b>	Resource save failed

### Example

```

<update plugin="\$gamma-resource-plugin\" resource="vnf"
    key="vnf-name = $requests.vnf.vnf-name"
    pfx="requests.vnf">
    <parameter name="vnf-name"
        value="\$requests.cust-country-code + $requests.cust-id + $requests.cust-city + $requests.cust-
state + '001VCE'" />
    <parameter name="vnf-type" value="vce" />
    <parameter name="orchestration-status" value="pending-create" />
    <parameter name="heat-stack-id" value="\$requests.heat-stack-id" />
    <parameter name="mso-catalog-key" value="\$requests.mso-catalog-key" />
    <parameter name="availability-zone"
        value="\$aic-availability-zone.availability-zone" />
    <parameter name="aic-site-id" value="\$requests.vnf.aic-site-id" />
    <parameter name="oam-ipv4-address" value="\$vce-ipv4-oam-addr.ipv4-addr" />
</update>

```