

# Mounting an O-RAN Managed Function

- [PNF Registration to VES Collector](#)
- [PNF Registration to Message Router \(DMaaP\)](#)
- [MountPoint creation on OpenDaylight's MDSAL](#)
- [MountPoint deletion on OpenDaylight's MDSAL](#)

There are several ways to "mount" an O-RAN Managed Function (also called O-RAN component) to an ONAP based SMO for O1 interfacing. The term "mount" means to configure the SMO and/or NetConf-Server of the O-RAN Managed Function to establish a NetConf Session.

According to O-RAN OAM Interface specification the [pnfRegistration](#) VES message is used. The VES message informs the SMO about a new aor again functional Physical Network Function (PNF, Device, ...). The concept could also be used for Virtual NetWork Functions (VNF).

For test purposes the pnfRegistration message content can be also send directly to the SMO Message Router (ONAP DMaaP) or in case of an OpenDaylight based O1-Controller a RestConf message can be send.

## PNF Registration to VES Collector

The PNF Registration content is defined by the ONAP project: [https://docs.onap.org/en/frankfurt/submodules/vnfrqts/requirements.git/docs/Chapter8/ves7\\_1spec.html?highlight=ves](https://docs.onap.org/en/frankfurt/submodules/vnfrqts/requirements.git/docs/Chapter8/ves7_1spec.html?highlight=ves) api#pnfregistration-domain-datatypes

However a couple of information are missing to create a NetConf connection:

- NetConf port
- Security information
  - user name
  - password
  - public key

To transport such data the HashMap 'additionalFields' is used.

## PNF Registration to Message Router (DMaaP)

text

## MountPoint creation on OpenDaylight's MDSAL

Please see the OpenDaylight Neon Documentation: <https://docs.opendaylight.org/projects/netconf/en/stable-neon/user-guide.html?highlight=mounting#spawning-new-netconf-connectors>

```
# Script to send a create mount-point request to OpenDaylight

# update the following nine parameters
protocol=https
controller=<controller-host>
port=443
basicAuth="<controller-user>:<controller-password>"

nodeId=identifier-of-network-function-at-smo
oamNodeIp=10.10.10.10
oamNodePort=830
oamNodeUser=netconf
oamNodePassword=netconf

# dont change blow
content=Content-Type:application/json
accept=Accept:application/json
path=restconf/config/network-topology:network-topology/topology
/topology-netconf
uri=$protocol://$controller:$port/$path/node/$nodeId

body='{ "node": [{ "node-id": "'$nodeId'", "netconf-node-topology:
host": "'$oamNodeIp'", "netconf-node-topology:port": "'$oamNodePort'", "
netconf-node-topology:username": "'$oamNodeUser'", "netconf-node-topology:
password": "'$oamNodePassword'", "netconf-node-topology:reconnect-on-
changed-schema": false, "netconf-node-topology:sleep-factor": 1.5, "netconf-
node-topology:tcp-only": false, "netconf-node-topology:connection-timeout-
millis": 20000, "netconf-node-topology:max-connection-attempts": 100, "netconf-
node-topology:between-attempts-timeout-millis": 2000, "netconf-node-topology:
keepalive-delay": 120} ] }'

curl -i -k -u $basicAuth -H $content -H $accept -X PUT -d $body $uri
```

## MountPoint deletion on OpenDaylight's MDSAL

```
# Script to send a delete mount-point request to OpenDaylight

# update the following nine parameters
protocol=https
controller=<controller-host>
port=443
basicAuth="<controller-user>:<controller-password>"

nodeId=identifier-of-network-function-at-smo

# dont change blow
content=Content-Type:application/json
accept=Accept:application/json
path=restconf/config/network-topology:network-topology/topology
/topology-netconf
uri=$protocol://$controller:$port/$path/node/$nodeId

curl -i -k -u $basicAuth -H $content -H $accept -X DELETE $uri
```