Common Node Properties

A number of Opmantek products use a common node configuration infrastructure, which supports standard, product-specific and custom node attributes. This page describes both the core properties and outlines the product-specific ones.

Common Properties

These are used by all Opmantek products (exception: NMIS does not use the `addresses` property). Only the first two, **name** and **host**, are **absolutely necessary** properties.

The properties shown in *italic* are not directly interpreted by the applications and are completely optional.

Property Name	Description
name	The internal name of the node. This is used for identifying and displaying the node, not for communication with the node!
host	The DNS name, unqualified host name or main IP address for this node. Using a fully qualified DNS host name is recommended.
addresses	A list containing any secondary IP addresses that should also be associated with this node. This list is often empty. NMIS does not use this property.
group	The (single) group that this node belongs to, which is used to control what nodes a user is allowed to see.
notes	Free-form notes for a node, which are shown on the relevant context pages. (optional)
comments	A list of originator- and time-tagged comments for this node. (optional, supported in opEvents 2.0 and newer)
location	The location of the node. (optional)
businessService	The service the node provides. (optional)

Common Properties for NMIS 9

For the new product generation, opEvents and opConfig use the same collection where NMIS 9 saves this data.

The properties shown in *italic* are not directly interpreted by the applications and are completely optional.

Property Name	Description
uuid	A unique identifier of the node
cluster_id	The identifier for the server. Indicates that this is a local node when it is the same as the cluster_id specified in the NMIS 9 config file.
configuration.host	The DNS name, unqualified host name or main IP address for this node. Using a fully qualified DNS host name is recommended.
configuration.addresses	A list containing any secondary IP addresses that should also be associated with this node. This list is often empty. NMIS does not use this property.
configuration.group	The (single) group that this node belongs to, which is used to control what nodes a user is allowed to see.
configuration.notes	Free-form notes for a node, which are shown on the relevant context pages. (optional)
configuration.comments	A list of originator- and time-tagged comments for this node. (optional, supported in opEvents 2.0 and newer)
configuration.location	The location of the node. (optional)
configuration. businessService	The service the node provides. (optional)

Properties for Licensing and Activation

Both opConfig (2.1 and up) and opEvents (1.2.3 and up) support per-node activation and licensing, which is controlled by the properties described below.

If a node is disabled for a particular product, then it is **completely ignored by this product**: for opEvents that means no events are processed for this node, for opConfig no command sets would be run for the node and so on.

By default nodes are enabled for all products except opConfig. Nodes are disabled if and only if they are explicitly set to disabled.

Property Name	Description
activated.opConfig	If set to 0, the node is disabled for opConfig. If unset or set to any non-zero value, the node is considered active.
activated.opEvents	If set to 0, the node is disabled for opEvents. If unset or set to any non-zero value, the node is considered active.
activated.NMIS	If set to 0, the node is disabled for NMIS. If unset or set to any non-zero value, the node is considered active. All nodes that are activated for NMIS are activated and licensed for opCharts.

Note: The activation properties are within a subhash/subdocument, and the listing above uses MongoDB dot-notation to indicate that (just like you would access such properties in an opEvents policy rule or an opConfig compliance rule).

opConfig Properties

opConfig requires two extra sets of properties for correct communication with a node, which are stored in the connection_info and os_info subhash /subdocuments. The os_info information is primarily used in command set definitions, and customized command sets might not use any of these properties.

The properties given in *italic* are optional and only relevant for specific types of devices.

Property Name	Description
connection_info.personality	The type of OS or system 'personality', which is used for determining what command sets are applicable.
connection_info.credential_set	Which credential set to use when authenticating to this node.
connection_info.transport	Which transport mechanism to use for connecting to this node. Valid choices are "SSH" or "Telnet".
connection_info.privileged_paging	Whether setting up paging requires privileged mode. Only relevant on Cisco PIX/ASA.
connection_info.line_endings	What line separator character(s) to send to the node. By default the patform's 'newline' character sequence is used.
connection_info. default_continuation	The name of the phrasebook macro to use for line continuation.
connection_info.connect_options	Any extra, non-standard options for the connection. See the documentation for Net::CLI::Interact for details. If present, this must be an Array in JSON format.
os_info.os	The general OS type. Required to determine what command sets are applicable.
os_info.version	The OS Version identifier.
os_info.major	The Major Release Number extracted from the OS Version.
os_info.train	The Train of the OS version. Relevant for Cisco devices only.
os_info.platform	The OS Platform as extracted from the Version.
os_info.image	The OS Image. Relevant mainly for Cisco devices.
os_info.featureset	The enabled Feature Sets of this system. Relevant mainly for Cisco devices.

OPCONFIG 4 This properties are accessible in the configuration hash. As an example, platform should be accessed as *configuration.os_info.platform*.

opEvents Properties

The opEvents does have a product-specific node property: **activated.opEvents**

NMIS Properties

NMIS uses further properties to control its operations, mostly things like SNMP-related identities, node active and collect flags, etc. The table below describes the most important properties that NMIS interprets or understands.

Whenever a node in opConfig or opEvents is refreshed/imported from NMIS, all of NMIS' node properties are also transferred. These properties are currently not used by opConfig or opEvents, but will show up when you export a node using opnode_admin. Your opEvents and opConfig policies can make use of these properties, you can edit them with opnode_admin (and, to some extent, via the GUI if you modify the configuration variables `opevents_gui_node_summary_list` and/or `opconfig_gui_node_summary_list`).

Property Name	Description
active	whether the node is active and handled by NMIS or should be ignored
collect	whether SNMP data should be collected from this node
ping	whether reachability statistics should be collected for this node
model	what type of device this is (default: "automatic")
version	snmpv1, snmpv2c or snmpv3; relevant only if collect is true
community	read-only community string for access using snmpv1 or snmpv2c
netType	wan or lan
roleType	core, access or distribution - the three layers of the Cisco three-layer hierarchical network model
username authprotocol authpassword privprotocol privpassword	authentication and privacy parameters for SNMP Version 3
authkey privkey	Alternative password parameters for SNMP Version 3. Optionally, the **authkey** and **privkey** can be used so that a plain text password does not have to be specified (see Net::SNMP). The `snmpkey` utility can be used to create the hexadecimal key string with the authoritativeEngineID (MAC address) for the destination device (see snmpkey).
port	which port to use for SNMP access (default: 161)
timezone	which timezone this system is in, numeric offset
webserver	whether the node runs a webserver. if true, a link to the node is presented in the NMIS GUI
threshold	whether thresholds are to be processed for this node
cbqos	whether Quality of Service data should be collected for this node (if the device and model support QoS)

NMIS 9 This properties are accessible in the configuration hash. As an example, collect should be query as *configuration.collect*.