Node Tags in YANG Modules

draft-ietf-netmod-node-tags-08
Qin WU (Huawei)
Benoit Claise(Huawei)
Peng Liu (CMCC)
Zongpeng Du (CMCC)
Mohamed Boucadair (Orange)
Recap

• Node tags classify nodes from different YANG modules and identifying their characteristic data.
  • The nodes can be node in the schema tree or instances of nodes in the data tree
• Use of Node tags is to search discrete categories of YANG nodes that are scattered across the same or different YANG modules supported by a device.

• This draft has passed through YANG Doctor Last Call Review and the first WGLC was initiated on April 8
  • Thanks for Jurgen, Balazs, Joe, Adrian, Qiufang, Jan, Andy for valuable comments
  • One open issue raised is whether the mechanism defined in this document generic enough to convey the meta-data information or specific use case driven?
• A few suggestions we received from Jurgen are:
  • Distinguish tag for nodes in the schema tree and tags for nodes in the data tree
  • Frequency of these tag changes
  • Tag retrieval scale
  • How tag are retrieved?
  • If I have YANG extensions to define tags at the module design time, why expose these tags vis instance data
  • How does a model writer decide using the RFC7952 mechanism and using this new mechanism
• V-08 has been issued to address comments received from WGLC and additional ones raised by Jurgen
Change 06-08

• Change the title into Node Tags in YANG Modules.
• Make objective clearly, cover tags for both nodes in the schema tree and nodes in the data tree.
• Distinguish Instance level tag from Metadata annotation in the introduction section.
• Consolidate opm-tag extension, metric-type extension and multi-source-tag extension into one generic yang extension.
• Remove object tag and property tag.
• Document clearly which tags can be cached and how applications are supposed to resynchronize and pull in any update in section 3.
• Clarify Instance level tag is not used to guide retrieval operations in section 3.
• Distinguish Schema Level tag from Instance level tag in the introduction section and section 3.
• Schema Level tag used in xpath query has be clarified in section 3.
• Update Model Tag design in section 5.1 based on Balazs's comments.
• Add Instance level tunnel tagging example in the Appendix.
• Add ‘type’ parameter in the base model and add one more model extension example in the Appendix.
Q1: schema node tag vs data node instance tag

• Question: - Do we talk about tags for nodes in the schema tree or do we talk about tags for nodes in the data tree?

• Answer:
  • Support both schema level node tags and instance level node tags
    • Originally focus on schema level node tags
    • Has expanded to support Instance node tags based on community interests and feedback
  • The node id in ietf-node-tags YANG module is defined as nacm:node-instance-identifier
  • Update abstract, introduction section and sample use case section in v-08 to distinct schema tree node tags from data tree node tags?

• Anyone objection to support both?
Q2: Frequency of these tag changes

• Question: What are our expectations regarding the frequency with which these tags change?

• Answer:
  • Similar to RFC8819, support both system tag and user configured tag
  • System will add tags into data node list of ietf-data-node-tags module if model writer uses YANG extensions to define tags at module design time (Schema level tag)
    • tag a set of specific schema nodes.
    • tags are static, will not change during the run time of the device.
  • The client can dynamically define or remove any tag on specific data node instance during the run time of the device, (instance level tag)
    • it’s possible for tags to be changed (i.e., added/removed), but not frequently
    • One client adds tag into node list of ietf-data-node-tags, tags are stored in the <operational>
    • another client subscribe to the tag changes from <operational>, any tag changes in the update will be resync to this client
  • See introduction section and sample use case section for more details
Q3: How tag are retrieved?

• Question: What are our expectations how tags are retrieved? Is tag retrieval associated with data retrieval or is tag retrieval largely detached from data retrieval?

• Answer:
  • tag retrieval is largely detached from data retrieval.
  • Tags will only be returned when the client retrieves node-tags data tree.
  • `<get-schema>` operation can also be used to discover the schema node tags defined during the module definition.
    • This is aligned with RFC8819 YANG Module Tags.
Q4: Tag retrieval scale

- Question: How does tag retrieval scale? What are common ways for retrieving tags? Is it required or desirable to use tags in filter operations?

- Answer:
  - use `<get>`/<get-config>/<get-data> for NETCONF protocol to retrieve tags.
    - General XPTAH-filter or subtree-filter can be used as filter operations.
    - No intentions to extend filter operations for tags retrieval other than standard mechanisms.
  - For schema level node tag, system add only one entry in the node list of ietf-data-node-tags module
  - For instance level node tags, client use edit-config to add tag or remove tags
    - The number of tag he managed is controlled by the client itself.
    - Suppose all instances of a given data node are marked with the different tags, these tags by design can't be used to retrieve every foo instance that is tagged bar.
Q5: Model Writer Guidance

• Question: How does a model writer decide between using the RFC 7952 metadata annotation mechanism and using this new mechanism?

• Answer:
  • Metadata annotation is tied with a given data node instances,
    • The value of the metadata annotation is decided and assigned by the server and sent to the client,
    • e.g., the origin value, indicates to the client the origin of a particular data node instance.
  • node-tags is retrieved centrally via the ietf-data-node-tags module.
    • node-tags is created and removed by the system client.
    • tag nodes in the schema tree, e.g., for a whole list.

• Question: How does a model writer decide between using this new mechanism and simple YANG extension statements for things that are static?

• Answer:
  • Instance level tag, tag the data node rather than the data module or other statement in the data module,
    • e.g., revision in the data module
    • Can provide data node auxiliary information or data node properties.
  • Schema level tag, can be deleted via the ietf-data-node-tags module and remove from operational datastore.
    But such tag still exist in the YANG data module.
Next Step

• Resolve all comments raised from the meeting
• Look for Guidance from the WG