List Pagination Mechanisms for NETCONF and RESTCONF

draft-wwlh-netconf-list-pagination-00
draft-wwlh-netconf-list-pagination-nc-02
draft-wwlh-netconf-list-pagination-rc-02

Kent Watsen
Qin Wu (Presenting)
Olof Hagsand
Hongwei Li
Per Andersson

NETCONF WG
IETF 112 (Virtual)
Recap

• Motivation and Goal:
  • To better support user-facing client interfaces to retrieve large number of entries from lists or leaf-lists.
  • Applicable to both config data and state data.
  • Examples: traffic logs, interfaces, ACLs, etc.

• Server-side processing reduces latency, bandwidth, and client-resources.

• Enables servers to leverage “indexes” maintained by their backend storage system

• Originally presented in IETF 109
  • 50+ number of msgs on RESTCONF Collection problem
  • List Pagination design team was set up.
List Pagination Acronyms

• List Pagination (LP):
  • A standard mechanisms to control the filtering, sorting, and retrieval of entries of list or leaf-list.

• NETCONF List Pagination (LP-NC):
  • NETCONF Extensions to Support List Pagination

• RESTCONF List Pagination (LP-RC):
  • RESTCONF Extensions to Support List Pagination
Since IETF 109

• **List Pagination (v00)**

  • Factored out from the "list-pagination-nc" and the "list-pagination-rc" drafts.

  • Renamed "count" and "skip" parameters into "limit" and "offset".

• General definitions

  • Five query parameters for list pagination.
  • One query parameter for descendant list pagination.
  • Per schema-node tags on server ‘list pagination’ capability constraints.

• Appendix contains example YANG module, data set, and queries.
Since IETF 109

• **List Pagination for NETCONF**
  
  • Change from new RPC "get-pageable-list" to augmenting three NETCONF "rpc" statements: get, get-config, and get-data.
  
  • Use grouping factored out from ietf-list-pagination in ietf-list-pagination-nc.
  
  • Provide example of list pagination with all query parameters in LP NC draft.

• **List Pagination for RESTCONF**
  
  • Align query parameters for RC with 6 query parameters for list pagination draft.
  
  • Declaring "list" and "leaf-list" as valid resource targets for the GET operation and optionally for the DELETE operation if needed.
  
  • Remove YANG module for LP RC.
List or Leaf-list Pagination

- **Query Parameters:**

<table>
<thead>
<tr>
<th>Query Parameters</th>
<th>Target Data object</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>where</td>
<td>List or leaf-list</td>
<td>Filter expression for result set entries to match</td>
</tr>
<tr>
<td>Sort-by</td>
<td>List or leaf-list</td>
<td>Node to sort result set entries</td>
</tr>
<tr>
<td>Direction</td>
<td>List or leaf-list</td>
<td>The direction of result set entries to be returned</td>
</tr>
<tr>
<td>Offset</td>
<td>List or leaf-list</td>
<td>The number of entries to be skipped in the result set</td>
</tr>
<tr>
<td>Limit</td>
<td>List or leaf-list</td>
<td>The number of entries returned</td>
</tr>
<tr>
<td>Sublist-limit</td>
<td>Descendant List or leaf-list</td>
<td>The number of entries returned   For descendant list or leaf-list, the target can be any node or a datastore</td>
</tr>
</tbody>
</table>

- **Server Processing order:**

  Where > Sort-by > Direction > Offset > Limit.

- **MetaData attribute - remaining:**

  - Used together with limit or sublimit.
  - Return the number of entries not included due to the limit operation.
List or Leaf-list Pagination Example

Combination of 6 query parameters used in the Request/Response message pair to guide how list or leaf-list should be returned.

A.3.7 of draft-wwlh-netconf-list-pagination
Descendant List or Leaf-list pagination

'Remaining' Usage Examples

Example 1: Target node: `/example-social:members/member=alice` in the `<intended>` datastore.

Example 2: Target node: `<intended>` datastore

---

A.3.6.1 of draft-wwlh-netconf-list-pagination

A.3.6.2 of draft-wwlh-netconf-list-pagination
Server List Pagination Capability Discovery

- Constraints parameters Definition:
  - **Constrained**: indicate which "config false" list and/or "leaf-list" nodes are constrained.
  - **Indexed**: and, if so, which nodes may be used in "where" and "sort-by" expressions.

- Constraints parameters Extensibility:
  - There may be other constraints that need to be added in this module, e.g., not supporting 100% of the XPath 1.0 syntax or derived-from() and derived-from-or-self() not supported.

```xml
module: ietf-list-pagination

augment /sys:system-capabilities/sysc:datstore-capabilities
  /sysc:per-node-capabilities:
    +--ro constrained?    empty
    +--ro indexed?       empty
```

```xml
<system-capabilities
  xmlns="urn:ietf:params:xml:ns:yang:ietf-system-capabilities"
  xmlns:es="http://example.com/ns/example-social"
  xmlns:lp="urn:ietf:params:xml:ns:yang:ietf-list-pagination">
  <datstore-capabilities>
    <datstore>ds:operational</datstore>
    <per-node-capabilities>
      <node-selector>es:audit-logs/es:audit-log</node-selector>
      <lgp:constrained/>
    </per-node-capabilities>
    <per-node-capabilities>
      <node-selector>es:audit-logs/es:audit-log/es:timestamp</node-selector>
      <lgp:indexed/>
    </per-node-capabilities>
    <per-node-capabilities>
      <node-selector>es:audit-logs/es:audit-log/es:member-id</node-selector>
      <lgp:constrained/>
    </per-node-capabilities>
    <per-node-capabilities>
      <lgp:indexed/>
    </per-node-capabilities>
  </datstore-capabilities>
</system-capabilities>
```
LP NC Protocol Extension

Augment 3 NETCONF RPC statement <get>,<getconfig>,<get-data>. Aligned with LP draft for query parameters.

```plaintext
module: ietf-list-pagination-nc

augment /nc:get/nc:input:
  +---w list-pagination
    +----w where? union
    +----w sort-by? union
    +----w direction? enumeration
    +----w offset? uint32
    +----w limit? union
    +----w sublist-limit? union

augment /nc:get-config/nc:input:
  +---w list-pagination
    +----w where? union
    +----w sort-by? union
    +----w direction? enumeration
    +----w offset? uint32
    +----w limit? union
    +----w sublist-limit? union

augment /ncds:get-data/ncds:input:
  +---w list-pagination
    +----w where? union
    +----w sort-by? union
    +----w direction? enumeration
    +----w offset? uint32
    +----w limit? union
    +----w sublist-limit? union
```
List Pagination NC with all parameters in the Request/Response message pair.

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="42">
  <get-config>
    <source>
      <running/>
    </source>
    <filter type="xpath" select="/es:members/es:member" xmlns:es="http://example.com/ns/example-social"/>
    <list-pagination xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-list-pagination">
      <where>\states\joined[starts-with(@timestamp,'2020')]</where>
      <sort-by>timestamp</sort-by>
      <direction>backwards</direction>
      <offset>2</offset>
      <limit>2</limit>
      <sublist-limit>1</sublist-limit>
    </list-pagination>
  </get-config>
</rpc>
```
draft-wwlh-netconf-list-pagination-rc-02
Update RESTCONF to Support List Pagination

3 Protocol Extensions:
1. Add list and leaf-list as valid resource target for the GET and DELETE operations.
2. Add new media type "application/yang-data+xml-list".
3. Add 6 new query parameters "limit", "offset", "direction", "sort", "where", and "sublist-limit".

<table>
<thead>
<tr>
<th>Name</th>
<th>Methods</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>limit</td>
<td>GET, HEAD</td>
<td>Limits the number of entries returned. If not specified, the number of entries that may be returned is unbounded.</td>
</tr>
<tr>
<td>offset</td>
<td>GET, HEAD</td>
<td>Indicates the number of entries in the result set that should be skipped over when preparing the response. If not specified, then no entries in the result set are skipped.</td>
</tr>
<tr>
<td>direction</td>
<td>GET, HEAD</td>
<td>Indicates the direction that the result set is to be traversed. If not specified, then the result set is traversed in the &quot;forwards&quot; direction.</td>
</tr>
<tr>
<td>sort-by</td>
<td>GET, HEAD</td>
<td>Indicates the node name that the result set should be sorted by. If not specified, then the result set’s default order is used, per YANG’s &quot;ordered-by&quot; statement.</td>
</tr>
<tr>
<td>where</td>
<td>GET, HEAD</td>
<td>Specifies a filter expression that result set entries must match. If not specified, then no entries are filtered from the result set.</td>
</tr>
<tr>
<td>sublist-limit</td>
<td>GET, HEAD</td>
<td>Limits the number of entries returned for descendant lists and leaf-lists. If not specified, the number of entries that may be returned is unbounded.</td>
</tr>
</tbody>
</table>

Deletion of a list

```
DELETE /restconf/ds/ietf-datastores:running/example-social:members/m\ember=bob/favorites/decimal64-numbers HTTP/1.1
Host: example.com
Accept: application/yang-data+xml
```

Response from the RESTCONF server:

HTTP/1.1 204 No Content
Date: Thu, 26 Jan 2017 20:56:30 GMT
Server: example-server

C.2 of draft-wwlh-netconf-list-pagination-rc
Open Issue 1: Cursors support

• For server w/ cursor support:
  • Enable the paging to continue over the snapshot, despite the dataset changing.
  • The client assumes the list doesn’t change??

• For server w/o cursor or snapshot support:
  • Might return larger remaining value than previous fetch during refetching page in a time series log (where logs are appended).

• For config, many systems have an internal reader/writer mutex, so a cursor would never be needed
  • Etags/Timestamps can be used to detect dynamic list change.
  • The error message is used to indicate the end of the list.

• Question: Do we have a compelling use case for this?
Open Issue 2: Zero vs Unknown

- The ‘remaining’ annotation is used to return the number of elements not included in a result set after a 'limit' or 'sublist-limit' operation.

- The current draft states that:

  “If no elements were removed, this annotation MUST NOT appear. The minimum value (0), which never occurs in normal operation, is reserved to represent 'unknown'.”

- But one best practice proposed by one of design team members is:
  - No more entries  --->  Return zero
  - Unknown        --->  Not return the "remaining" annotation at all

- Comments and suggestions?
Next Step

• Two-partial implementations are underway
  • One from Kent;
  • The other from Olof.

• Questions, comments?
Thank You