

Common YANG Data Types

draft-ietf-netmod-rfc6991-bis-04

Jürgen Schönwälder
Jacobs University Bremen

rfc6991bis: yang:date

- XML schema seems to use a different canonical format, we need to take a closer look how to define the canonical format given that a date really identifies a 24 hour interval and what XSD means with 'interval midpoint'.
- Shall we make the timezone offset optional? The timezone offset is optional in XML schema. There is CBOR work in progress to define a date without a timezone. The downside of optional timezones is that two systems in different timezones may assign different dates to the same event. On the other hand, there may be use cases where the timezone is implicit.
- Proposal: Make the timezone optional (adding text describing the implications, i.e., comparisons may be surprising). Align as much as possible with the XML schema definition concerning the canonical format.

rfc6991bis: yang:xpath1.0

- How do we deal with xpath expressions in other encodings, such as JSON? Do we assume an xpath context populated with module names such that module names can be used to qualify path expressions. This may need discussion and/or a new definition.
- This interacts with the definition of node-instance-identifier.
- Options: (i) Leave this definition as it is. (ii) Detail how this type works with encodings that use module names instead of prefixes to qualify names.
- Proposal: Leave this definition as it is.

rfc6991bis: yang:node-instance-identifier

- This is taken from RFC 8341 and the idea is that this definition is useful without requiring a dependency on NACM.
- What does the second bullet actually do? Do we keep this? ("The set of variable bindings contains one variable, 'USER', which contains the name of the user of the current session.")
- This interacts with the definition of xpath1.0 concerning the context and the use of module names as prefixes.
- Options: (i) Do not include node-instance-identifier, people can use the NACM version for now. (ii) Include node-instance-identifier and detail how this type works with encodings that use module names instead of prefixes to qualify names. (iii) Introduce a new node-instance-identifier type based on RFC 8040 section 3.5.3 that always uses module names and no prefixes.
- Proposal: Do nothing now (i) and implement (iii) when YANG 2.0 removes XML specifics such as prefixes from the YANG core.

rfc6991bis: yang:longitude, yang:latitude, yang:postal-code, yang:country-code

- It was suggested to add types for longitude, latitude, postal code, country-code. Do we go there or do we leave these for other modules to define? It seems some definitions should go into draft-ietf-netmod-geo-location.
- For country codes, there is ISO 3166, which defines two-letter, three-letter, and numeric country codes. I assume people wanted two-letter codes (as used in the DNS), i.e., they want DE and not DEU. But note that it is GB and not UK, i.e., what we commonly use in the DNS may not be exactly ISO 3166. (The devil is always in the details.)
- For postal codes, it is unclear what the requirements are or what a proper definition for postal codes is. It is not entirely clear what the authoritative definition of the format of postal codes is, perhaps the Universal Postal Union.
- Options: (i) do nothing since there is draft-ietf-netmod-geo-location or (ii) add a country code definition only or (iii) add both a country code definition and a postal code definition (which might be to some extent vague)
- Proposal: Do nothing

rfc6991bis: yang:percentage

- Percentages are frequently used in YANG models but usages differ a lot in precision and range. It is not clear what the proper generic definition of a percentage type would be and whether it is worth having it.
- RFC 7950: `typedef percent { type uint8 { range "0 .. 100"; } }`
- RFC 8294: `typedef percentage { type uint8 { range "0..100"; } }`
- I-Ds: `typedef percentage { type decimal64 { fraction-digits 5; } }`
`typedef percentile { type decimal64 { fraction-digits 2; } }`
- Proposal: Do not add a percentage type since it is trivial to define a context specific percentage type that matches range and precision requirements (and there is already a definition in RFC 8294 for those who need exactly that definition).

rfc6991bis: inet:host

- Lada suggested to replace the inet:domain-name usage in the inet:host union with a new host-name definition that follows the NR-LDH definition in RFC 5890.
 - The goal is (i) to disallow single letter host names and (ii) to disallow the underscore character in host names and (iii) to restrict host names to non-reserved letter-digit-hyphen (NR-LDH) labels.
 - typedef host-name {
 - type domain-name {
 - pattern '[a-zA-Z0-9\-\.\.]+';
 - length "2..max";
 - }
 - description
 - "Host names must be at least two characters long (see RFC 952) and they are restricted to labels consisting of letters, digits and hyphens separated by dots (see RFC1123 and RFC 952).";
 - reference
 - "RFC 952: DoD Internet Host Table Specification
 - RFC 1123: Requirements for Internet Hosts: Application and Support";
 - }
- Proposal: If we can reach strong agreement on a host-name typedef, add it and change the inet:host union to use it. If we do not reach agreement, leave things unchanged.

rfc6991bis: inet:email-address

- Current definition follows addr-spec (idea: follow mailto: url scheme format).
- The current pattern does not take care of quoted-string, obs-local-part, domain-literal, obs-domain. Do we need all of these? Perhaps best to support them but this makes tight regular expressions non-trivial.
- Proposal: Add the typedef covering addr-spec but this work would benefit from some help to get the pattern right. Any volunteers to write/review test cases?

rfc6991bis: inet:urn-scheme, inet:uri-authority, inet:uri-path, inet:uri-query, ...

- It is not clear how commonly useful these types are, the WG was pretty silent about this proposal.
- On the technical side, it is unclear whether data is represented with percent escapes resolved or not.
- Proposal: Do not add these typedefs since their use seems rather specific

rfc6991bis: loopback addresses

- There was a request to add types for loopback addresses (127.0.0.0/8 and ::1/128).
- It is not clear how commonly useful these types are, these types seem to be of rather limited value.
- This is related to an effort to define a YANG module for MPLS LSP Ping (RFC 8029) but the details are unclear, i.e., what is exactly needed and how such a type will be used and whether there is a common need for types for loopback addresses.
- Proposal: Do not add these types at this point in time