Network Modeling
Internet-Draft
Intended status: Informational
Expires: 18 September 2026

R. Wilton
Cisco

17 March 2026

YANG Next Agreement
draft-wilton-netmod-yang-next-agreement-00

Abstract

The purpose of this document is to discuss and hopefully find agreement on the scope and shape of the next version of YANG.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://rgwilton.github.io/yang-next-agreement/draft-wilton-netmod-yang-next-agreement.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-wilton-netmod-yang-next-agreement/.

Discussion of this document takes place on the Network Modeling Working Group mailing list (mailto:netmod@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/netmod/. Subscribe at https://www.ietf.org/mailman/listinfo/netmod/.

Source for this draft and an issue tracker can be found at https://github.com/rgwilton/yang-next-agreement.

Status of This Memo

Table of Contents

1. Introduction

The YANG Next Issue Tracker (https://github.com/netmod-wg/yang-next/issues/) lists around 125 open issues and 35 closed (some perhaps closed because they would have been non-backwards-compatible changes which were not be considered at the time that they were being evaluated).

These issues are of varying size, complexity and importance. If the contents of the next version of YANG is defined solely by which issues folks would like to work on (i.e., the bazaar method) then there is a risk that we will end up with a potential large and somewhat inconsistent update to the language. It is unclear whether this would gain consensus within the IETF or widespread traction within the industry.

Hence, this document proposes that attempt to reach clear consensus on what problems a revised version of YANG is intending to solve and for the issues to be categories into 4 sets:

1. issues that must be included in a next version of YANG

2. issues that should be consider for the next version of YANG (if there is sufficient interest)

3. issues that should not be considered for the next version of YANG, but could be considered for future versions.

4. issues that should not be considered further, i.e., implementing them would likely be too harmful for the language because it would make the language too big or complex or the specification would be too long.

It is worth noting that a couple of separate efforts to score issues has been made, e.g., adding meta-data annotations, or closing the issues, but that in itself does not indicate what issues should be worked on.

In addition, it should be noted that Appendix "Issue 152" has proposed a slightly different scoring/classification of issues (by Andy Bierman).

There is no goal to publish this document as an RFC, and probably the issues could be tracked via a github dashboard. But a document like this may be a good way to ensure that the working group is aware of what changes are proposed and to ensure that there is consensus.

2. Proposed changes that should be made to YANG

The set of issues in this section are ones that the authors believe should be added to any future version of YANG. Many of these issues are clarifications, small/easy additions, or high importance issues.

Summary of proposed changes:

* Cleanup up the base specification:
  - Factor out XML encoding rules, Appendix "Issue 10"
  - Move NETCONF specifics out, Appendix "Issue 11"
  - Remove normative references to RFC 6241, Appendix "Issue 12"
  - Apply verified errata, Appendix "Issue 134"
  - Further restrict YANG Unicode to exclude "del" and C0 control characters, Appendix "Issue 125"
  - Is any NMDA (RFC 8342) cleanup needed (*no github tracking issue*)
* Apply clarifications, 13 issues listed in Section 2.1
* Deprecations & removals :
  - Deprecate "import by exact revision", Appendix "Issue 75"
  - Remove the "anyxml" statementAppendix "Issue 105"
* Fold in some keywords/functionality from extension drafts:
  - Incorporate structure extension, Appendix "Issue 8"
  - Deprecate "import by exact revision", Appendix "Issue 75"
  - Make NACM default-deny-all and default-deny-write built-in statements, but may need to generalize rather than being tied to NACM, Appendix "Issue 61"
  - YANG versioning, Appendix "Issue 45", Appendix "Issue 65", Appendix "Issue 66"
* 12 minor enhancement, in Section 2.2
* 13 larger improvements, in Section 2.3

In total, this is about 51 of the YANG Next issues that have been raised.

2.1. Clarifications

All of these issues are applying clarifications that are not intended to change the behaviour in the base specification, just make the specification clearer. For some of these issues, after review, the conclusion may be that no changes are needed.

Issue	Additional information	
Appendix "Issue 27" Clarify YANG "status" keyword usage (e.g., hierarchical)		
Appendix "Issue 44" Clarify if multiple deviations target the same schema parts		
Appendix "Issue 88" clarify NP-containers		
Appendix "Issue 96" Clarify definitions of YANG schema tree, vs module schema tree	This termology is confusing, we should cleanup/clarify	
Appendix "Issue 99" Clarify duplicate revision dates used in revision history		
Appendix "Issue 103" Clarify implicit ’case’ behavior		
Appendix "Issue 104" clarify "instance-required" behavior in typedefs		
Appendix "Issue 110" Clarify canonical order in RFC 7950		
Appendix "Issue 111" Clarify whether revision-dates must be unique or not		
Appendix "Issue 115" Clarify the meaning of properties which have		

	default value			
	Appendix "Issue 116" Clarify the behavior if the schema nodes in XPath expression are un-supported			
	Appendix "Issue 126" Injection of circular imports by deviation	Clarify rejection, or allow circular imports		
	Appendix "Issue 146"	Consistent default value behavior for operations and validation		

Table 1: Clarifications

2.2. Minor enhancements

This list of changes is for relatively minor/isolated enhancements. I.e., the changes are expected to be more isolated, or eady to add.

Issue	Consideration Reason
Appendix "Issue 5" default to namespace urn:yang:<module-name>	Removes unnecessary noise from the language.
Appendix "Issue 28" add 'status' as a sub statement to 'module'	Seems like small/easy addition
Appendix "Issue 33" Tag YANG identity as an intermediate base for classification only	Seems like a small useful adition
Appendix "Issue 34" Add native support for float/ double	We should just do this, Dec64 causes pain in specs and implementations
Appendix "Issue 59" Preliminary Status	Should be able to introduce unstable data nodes to mature models (e.g., "status experimental")
Appendix "Issue 94" deref() function for leafref statements	Implementations already allow this.

Appendix "Issue 98" Disallow if-feature statements where enabling the feature removes data nodes from the schema	We should just fix this
Appendix "Issue 101" allow 'require-instance' to be refined	Simple change
Appendix "Issue 125" Further restrict YANG Unicode to exclude "del" and C0 control characters	Same tweak to conform with RFC 9839
Appendix "Issue 128" Relax rules on usage of deprecated or obsolete identifiers	Possibly dup of Appendix "Issue 65" or Appendix "Issue 66"
Appendix "Issue 130" Add 'deprecated' statement	Extra meta-data that is easy to add.
Appendix "Issue 155" Allow hexadecimal notation for enum values	Seems like a small enhancement

Table 2: Minor Enhancements

2.3. Larger improvements

This list of issues tracks more important issues that will require more work, discussion, (and probably text) to specify but are deemed to be important to help grow the language.

Issue	Consideration Reason
Appendix "Issue 7" Support media-type specific schema that can be used to model error-info and other mount-points	YANG needs encoding agnostic way of returning errors.
Appendix "Issue 40" Allow deviation for Identities	E.g., to allow identities to be marked as not-supported

Appendix "Issue 49" Introduce critical extensions	Would be allowed to extend YANG semantics
Appendix "Issue 51" A general way to add stmts to any part of a schema	Would it very useful, but is a bigger change
Appendix "Issue 56" context-independent encoding of instance-identifiers and identityrefs	We should have a module name prefix based scheme
Appendix "Issue 62" Currently, list keys are all mandatory - allows default values for keys	Adding key defaults would simplify some model usecases
Appendix "Issue 70" Introduce support for critical annotations	Would be useful addition
Appendix "Issue 76" Make submodule "include" statement require a revision date.	Required to make module definition sound
Appendix "Issue 80" enable a server express conformance to a set of identifiers	Same as issue Appendix "Issue 40"
Appendix "Issue 95" Allow an module import to be defined as "types only"	Adds a useful refinement to import dependencies.
Appendix "Issue 121" Add support for hierarchical default values	These turn up in modeling and should be relatively easy to implement.
Appendix "Issue 149" Error statement for actions rpcs	Simplifies RPC/ protocol bindings, same/similar as Appendix "Issue 7"
Appendix "Issue 154" Add ability to remove nodes from a grouping in a "uses" statement	This would be helpful and improve reuse

Table 3: Larger Improvements

3. Proposed issues to consider for the next version of YANG

This list includes items that require further evaluation. Once fully evaluated all items should end up in one of the other lists, i.e., either we plan on making the change in the next version of YANG or we don't.

Issue	Consideration Reason
Appendix "Issue 2" Allow if-feature-stmt inside deviation-stmt	Too many platform files/ variants.
Appendix "Issue 4" Add a "map" statement	YANG list is unusal, but too late to change?
Appendix "Issue 13" Modify usage examples to be less NETCONF focused	Need to keep examples.
Appendix "Issue 16" Allow when in action	Relatively simple addition?
Appendix "Issue 21" Restrict regex to a subset of XML regex specification	Would make implementation easier, could use the IETF Regex spec?
Appendix "Issue 26" Consider removing support for sub modules from YANG	Proposal is to deprecate to simplify
Appendix "Issue 29" Clarify YANG validation for data nodes 'decoded' from anydata to a tree of real nodes	Was closed, but there is draft *TODO* related to this?
Appendix "Issue 30" key-predicate-expr should be (quoted-string / integer-value / decimal-value)	Was closed, but perhaps needs to be re-evaluated for a YANG 2.0?
Appendix "Issue 31" Add 'clone' statement	Was closed, but should consider reuse outside groupings?
Appendix "Issue 32" Allow Augmentation to Groupings	Was closed, but should we reconsider?
Appendix "Issue 36" enable leafrefs to uniquely reference a nested list	needs further investigation

Appendix "Issue 39" Allow deviation for description	Unclear whether this is really helpful/wise?
Appendix "Issue 41" Allow some references to from config-true to config-false (add capabilities)	Probably too big for next version of YANG, spec separately
Appendix "Issue 46" Binary encoding support (lets some types have binary persistence)	Should consider, discuss with CORE folks
Appendix "Issue 57" introduce if-module	Is this worth the effort?
Appendix "Issue 58" Introduce XPath function datastore()	Evaluate - Need to understand how this would be used?
Appendix "Issue 60" Allowing module private groupings, typedefs	Allow some top level definitions to be private to modules.
Appendix "Issue 63" Should it be possible to deviate "status"?	Should consider, e.g., to indicate obsolete nodes as still being supported?
Appendix "Issue 68" Add if-feature on must stmt (removed "on import stmt" part)	Should consider
Appendix "Issue 69" Clarify ’deviation’ substatements to match ABNF grammar	Helpful to clarify
Appendix "Issue 72" Introduce an annotation that resolves the union member	Can we get unions to be consistent across encodings?
Appendix "Issue 78" allow ’must’ as a sub statement to ’grouping’	Would need to better understand implications first
Appendix "Issue 81" let ’description’ be a substatement to ’input’ and ’output’	Not sure how important this is, but seems easy to add.
Appendix "Issue 83" Clarify canonical representation of typedefs	Clarifications are helpful.

Appendix "Issue 84" allow notifications/actions to appear in invalid contexts	Useful if not too hard to implement.
Appendix "Issue 87" enable specifying augmentation’s location amongst siblings	Was closed, but input/output are deemed to be ordered?
Appendix "Issue 90" Have a mechanism to allow enums to be extended	Should investigate and add if not too much complexity.
Appendix "Issue 92" enable if-feature statements to be "refined" into notifications and actions	We should consider adding this.
Appendix "Issue 97" enable ’ordered-by’ to be refined into a list or leaf-list	We hit this issue when supporting some OC models
Appendix "Issue 100" Add errata-stmt to YANG	Need to specify/agree a solution, but support fixing this
Appendix "Issue 106" Allow "input/output" to be defined without any child data nodes	Seems like a small change, but what about augment ordering?
Appendix "Issue 108" Allow when in notification	Probably should allow
Appendix "Issue 109" Change and compatibility rules for config=false (state) data	Should consider, but may be too much complexity
Appendix "Issue 122" Changing an identity base	This looks like an issue to consider addressing (but it might be rare)
Appendix "Issue 129" Enable reusability without groupings	Should investigate what this could look like but may just be too complex.
Appendix "Issue 131" make annotation a built-in YANG statement	Perhaps use critical extensions and keep out of the base language

Appendix "Issue 133" support for more efficient CBOR representation of strings	Potentially worth investigating
Appendix "Issue 138" Add new node type 'listref'	Proposing a better way of referencing lists. See also Appendix "Issue 77"
Appendix "Issue 139" Clarify adding mandatory nodes with augment + when	This might be worth relaxing/clarifying
Appendix "Issue 143" Allow deviation-stmt within uses-stmt	Proposal is allow more refinement.
Appendix "Issue 144" YANG 1.1 translation	Unclear if this would even be possible ...

Table 4: Possible issue for next YANG version

4. Open issues that should not be made for the next version of YANG

I think that the biggest challenges or potential long term enhancements to YANG are:

* replacing XPath with a YANG specific expression language.

* having a cleaner way of handling lists with multiple keys, e.g., when referenced

* a generic way to augment additional statements into a YANG module.

* more advanced or flexible reuse beyond groupings, including perhaps the ability to augment into a grouping.

Issue	Consideration Reason
Appendix "Issue 14" Allow deviations to modify "when" statements	Does this add to much complexity?
Appendix "Issue 15" Incorporate/ merge RFC 7952 (yang-metadata)	Not that widely used, better as a separate extension rather than in core document?
Appendix "Issue 18" add a	Done as a separate draft,

templating mechanism	wait for it to mature first
Appendix "Issue 19" yang canonical integer format	Not sure whether this is really needed.
Appendix "Issue 22" Restrict usage to a subset of XPATH	Better solution may be to define separate YANG expression langauge
Appendix "Issue 48" when using a grouping, the designer should be able to modify just about any aspect of the grouping	Was closed, may have too much accidental complexity?
Appendix "Issue 53" Create a way for a statement to tie-in with augment/deviation	Would seem to add a lot of complexity
Appendix "Issue 54" Define a way for extensions to declare sub-statement validity	Nice to have. Better to do in extension draft?
Appendix "Issue 55" define an encoding-independent "ypath:1.0" type	Was closed, but draft planned.
Appendix "Issue 71" extension-stmt conformance	More investigation might be needed, but might be too much work for base YANG spec.
Appendix "Issue 73" Initial value	Wanted by 3GPP, but complexity in what it really means? Does immutability work come into play here.
Appendix "Issue 74" Support for unique leaf (or leaf combo) in a list of lists	Adds complexity to implementations, is this really needed?
Appendix "Issue 77" Add support for a tuple type	Not necessarily a tuple type but would like a nicer solution for multi-keyed lists
Appendix "Issue 82" enable features to be supported per grouping-use (not globally per-	Too high complexity?

datastore)	
Appendix "Issue 86" allow 'case' as a substatement to 'grouping'	Not sure it is worth the complexity
Appendix "Issue 89" add 'recommended-default' statement	Not worth the additional complexity
Appendix "Issue 91" Consider relaxing identity uniqueness to only require uniqueness with the base identity hierarchy	Nice to have, but is it worth it?
Appendix "Issue 93" support 'dynamic default'	Some models need this, but should limit the complexity.
Appendix "Issue 113" Treat if-feature which references a non-existing feature as valid YANG but not enabled	Unsure whether this is really needed?
Appendix "Issue 119" The next step of XPath	We should do something here, but it needs to be a separate draft
Appendix "Issue 123" Allow identities to be active even when module is not implemented	I don't think that we need to change behaviour, but clarification might be needed/helpful.
Appendix "Issue 127" Relax rules for identityref representation without a prefix	Postpone to XPath replacement
Appendix "Issue 147" New default-system statement to indicate that the default value is not constant (determined by the system)	Seems complex, does system-datastore solve this anyway?
Appendix "Issue 148" YANG++	This probably shouldn't be YANG, but a new language
Appendix "Issue 150" nbc-change-stmt	I don't think that we should do this now
Appendix "Issue 153" Allow 'when' to be a child of the 'refine' statement	Not sure how this would work (without a container)

Appendix "Issue 156" canonical forms for strings (aka: the mac-address issue)	Need to spot data nodes that require custom handling.

Table 5: Possible issues for a future YANG version (not the next one)

5. Closed issues

This set of issues are either already closed or the author believes that they should be closed and should not be consider further for any future version of YANG.

Issues may be put in this section for any of these reasons:

* the issue is already marked as closed in github, not some closed issues are present on the previous lists because the authors believe that circumstances have changed (i.e., a YANG 2.0 version might be an option) and the issues should be reconsidered.

* the issue is a duplicate

* the issue is due to a misunderstanding of the YANG language or specification

* the consensus is that adding/introducing this issue would be harmful to the YANG language, perhaps taking it too far from its goals, or introducing too much complexity.

* this issue relates to the protocols not the language and hence the issue should be moved to a network protocol issue tracker.

5.1. Open issues proposed for closure in github

This list of issues are not currently marked as being closed in github, but the author is proposing that we close them, in that we _currently_ do not consider them for implementation in any YANG version.

Issues Proposed for Closure	Closure Justification
Appendix "Issue 6" Provide a correct ABNF for Yang strings	Perceived benefit is not worth the effort
Appendix "Issue 17" replace ’encoding’ with ’representation’	I’m not convinced this is really necessary

Appendix "Issue 102" ascii vs. unicode strings	Seems too low priority, should close (also need to fix link)
Appendix "Issue 107" Add deviate(not-supported) support for identities	Dup of Appendix "Issue 40", should close
Appendix "Issue 112" Support for conditional default values	Seems too complex, should close
Appendix "Issue 117" Add dynamic feature to YANG	Solution looks very complex.
Appendix "Issue 124" Add ability to deviate or change status of an identity	Should close, dup of Appendix "Issue 40"
Appendix "Issue 135" Make YANG Semver real statements instead of extensions	Dup of Appendix "Issue 45"
Appendix "Issue 136" Revise Module Update Rules	Should be covered by Appendix "Issue 45"
Appendix "Issue 140" grouping usage requirements	Propose closing, should be in rfc8407bis
Appendix "Issue 141" Add automatic list key generation (autokey)	Proposing closing, this is a protocol not YANG issue
Appendix "Issue 145" Let a presence container have a default presence?	Propose closing, not implementable
Appendix "Issue 151" YANG profiles and views	Too complicated, packages may help

Table 6: Currently open issues proposed for closure

5.2. Already closed issued in Github

This follow list of issues are those already marked as being closed in the YANG Next Github tracker.

Already Closed Issues	Closure Justification
Appendix "Issue 1" Only one idea per issue please!	Not a YANG issue (used for tracking only)
Appendix "Issue 3" Allow prefix statement to be optional for modules that don’t need it	Undesirable change
Appendix "Issue 9" Add an inactive metadata annotation	Protocol extension not a language feature
Appendix "Issue 20" add explicit module version-stmt	Duplicate of Appendix "Issue 45"
Appendix "Issue 23" add ’conformance-type’ leaf to ’import’ statement	Probably better done separately (e.g., packages)
Appendix "Issue 24" add a ’backwards-compatibility’ leaf to ’revision’ statement	Duplicate of Appendix "Issue 45"
Appendix "Issue 25" make yang more object-oriented	Too complicated - better discuss as part of Appendix "Issue 148"
Appendix "Issue 35" Simplify ’when’ statement processing	Closed, *TODO* possibly should be moved to be part of protocol spec?
Appendix "Issue 37" YANG ANBF could be defined in a simpler way	Not worth changing now
Appendix "Issue 38" Allow action to be invoked in the context of configuration datastore	Undesirable behaviour
Appendix "Issue 42" make schema-mount extension into a built-in statement	Critial extension might be a better choice
Appendix "Issue 43" Support for multiple key statements in a list	Too complex, other alternatives exist
Appendix "Issue 47" support external module requirements	Could be done in YANG library, packages.

Appendix "Issue 50" YANG Packages (multi-module conformance/guidance)	Separate draft, also not a language issue. But perhaps want to discuss conformance.
Appendix "Issue 52" add a statement for modeling checkbox dialogs	Better done witih extensions/overlay
Appendix "Issue 64" Clarify if double quotes are needed around identifiers	Deemed unnecessary
Appendix "Issue 67" clarify "require-instance" property for leafrefs	Deemed not to be an issue
Appendix "Issue 79" add ’unique’ as a substatement to ’leaf-list’	Not needed
Appendix "Issue 85" add "uses" as a sub-statement to "augment"	ALready supported in YANG 1.1
Appendix "Issue 114" Don’t treat non-exist import module as a error if there is no any effective reference to this import module in current module	Too hard to implement
Appendix "Issue 118" Co-existence between YANG1.0 and YANG1.1	Can’t change existing specifications.
Appendix "Issue 120" allow more restriction on Identity-ref	Closed as a duplicate of other issues
Appendix "Issue 132" Allow config=true list without a key-stmt	Closed as dup
Appendix "Issue 137" Add validation-rules-stmt to identify YANG validation scope	Closed, seems too complex
Appendix "Issue 142" Allow if-feature on deviations	Duplicate of Appendix "Issue 2"

Table 7: Already closed issues

6. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

7. Security Considerations

N/A. This document is only intended to help the WG reach consensus on the future direction of YANG and contains no protocol work.

8. IANA Considerations

This document has no IANA actions.

9. Acknowledgments

TODO acknowledge.

10. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

Appendix A. List of YANG Next Github Issues

This appendix summarizes issues from the YANG-next issue tracker. The summarization was created by LLM so it may not be entirely accurate.

Issue 1 (https://github.com/netmod-wg/yang-next/issues/1)

Only one idea per issue please!

Requested that issues be split so each issue covers one idea to ease prioritization and tracking. Closed as a process note rather than a YANG language issue.

* *Github Labels*: Not a YANG Issue

* *Status*: closed

Issue 2 (https://github.com/netmod-wg/yang-next/issues/2)

Allow if-feature-stmt inside deviation-stmt

Customers have complained that each platform/revision needs its own deviation file. They would like to define features matching platforms and have 1 deviations file per product family.

* *Github Labels*: complexity-med, backcompat-high, importance-med

* *Status*: open

Issue 3 (https://github.com/netmod-wg/yang-next/issues/3)

Allow prefix statement to be optional for modules that don’t need it

Proposed making the prefix statement optional when a module does not reference its own prefix. Objections noted parser ambiguity and the benefit of consistent prefixes across imports. Closed.

* *Github Labels*: Undesirable outcome

* *Status*: closed

Issue 4 (https://github.com/netmod-wg/yang-next/issues/4)

Add a "map" statement

From 9.2. YANG lists as maps YANG has two list constructs, the ’leaf-list’ which is similar to a list of scalars (arrays) in other programming languages, and the ’list’ which allows a keyed list of complex structures, where the key is also part of the data values.

* *Github Labels*: complexity-high, backcompat-high, importance-med

* *Status*: open

Issue 5 (https://github.com/netmod-wg/yang-next/issues/5)

default to namespace urn:yang:<module-name> ?

On 4/20/16, 7:47 AM, "netmod on behalf of Juergen Schoenwaelder" <netmod-bounces@ietf.org on behalf of j.schoenwaelder@jacobs-university.de> wrote: > On Tue, Apr 19, 2016 at 09:37:03PM +0200, Martin Bjorklund wrote: > > But I like urn:yang:<module-name> even more - and if we had this, we > wouldn’t have to use the "namespace" statement.

* *Github Labels*: complexity-med, backcompat-high, importance-low

* *Status*: open

Issue 6 (https://github.com/netmod-wg/yang-next/issues/6)

Provide a correct ABNF for Yang strings

(This is derived from a mailing list message, The issue that concerns me is that the ABNF doesn’t specify what is allowed as a string. I’m used to programming language definitions, where the grammar is specified quite rigidly, to the point that the ABNF can be input to a parser generator.

* *Github Labels*: complexity-low, backcompat-high, importance-low

* *Status*: open

Issue 7 (https://github.com/netmod-wg/yang-next/issues/7)

Support media-type specific schema that can be used to model error-info and other mount-points

The <error-info> element included in NETCONF and RESTCONF is very useful but there is no way to specify YANG schema to match expected data-model specific content. The ietf-restconf module defines the restconf-media-type extension that uses groupings to define this sort of data so they do not appear to be data nodes.

* *Github Labels*: complexity-med, backcompat-high, importance-high

* *Status*: open

Issue 8 (https://github.com/netmod-wg/yang-next/issues/8)

Incorporate/merge RESTCONF’s artifact extension (e.g. rc:yang-data)

The head of the on-list thread is here: But for some reason it doesn’t thread-in some responses: Note: the last response does seem to have threading working again, so be sure to check out other responses from others.

* *Github Labels*: complexity-low, backcompat-high, importance-low

* *Status*: open

Issue 9 (https://github.com/netmod-wg/yang-next/issues/9)

Add an inactive metadata annotation

The conditional-enablement draft seems to have support, but no one wants to rev NC/RC for it, so adding it into a rev of YANG might be the best option...

* *Github Labels*: importance-med, backcompat-unknown, complexity-unknown, Workaround is better

* *Status*: open

Issue 10 (https://github.com/netmod-wg/yang-next/issues/10)

Move normative XML encoding rules into its own RFC

The YANG RFC is littered with XML encoding rules (see table of contents). These should be factored out into another document like RFC 7951.

* *Github Labels*: complexity-med, backcompat-high, importance-med

* *Status*: open

Issue 11 (https://github.com/netmod-wg/yang-next/issues/11)

Move NETCONF-specific sections to NETCONF WG documents

The YANG RFC is contains sections entitled "NETCONF XML Encoding Rules" that don’t belong in the YANG spec. These should be moved to some future version of 6241, or another draft maintained by the NETCONF WG.

* *Github Labels*: complexity-med, backcompat-high, importance-med

* *Status*: open

Issue 12 (https://github.com/netmod-wg/yang-next/issues/12)

Remove normative references to RFC 6241

The YANG specification should not have any normative references to RFC 6241. Other than the references from the sections to be removed via issue #11, the only other substantive references are in the Terminology section, which references the following form RFC 6241: o configuration data o configuration datastore o datastore o state data These should probably be moved to the datastore RFC...

* *Github Labels*: complexity-low, backcompat-high, importance-med

* *Status*: open

Issue 13 (https://github.com/netmod-wg/yang-next/issues/13)

Modify usage examples to be less NETCONF focused

The Usage Example sections throughout the spec illustrate NETCONF usage. Either these examples should be updated to be equal parts NETCONF and RESTCONF, or they should be removed entirely.

* *Github Labels*: complexity-low, backcompat-high, importance-med

* *Status*: open

Issue 14 (https://github.com/netmod-wg/yang-next/issues/14)

Allow deviations to modify "when" statements

Should be self-explanatory; allow deviations to add, remove or replace when statements like they can with must statements.

* *Github Labels*: complexity-med, backcompat-high, importance-med

* *Status*: open

Issue 15 (https://github.com/netmod-wg/yang-next/issues/15)

Incorporate/merge RFC 7952 (yang-metadata)

RFC 7952 says: Due to the rules for YANG extensions (see Section 6.3.1 in), annotation definitions posit relatively weak conformance requirements. The alternative of introducing a new built-in YANG statement for defining annotations was considered, but it was seen as a major change to the language that is inappropriate for YANG 1.1, which was chartered as a maintenance revision.

* *Github Labels*: complexity-low, backcompat-high, importance-low

* *Status*: open

Issue 16 (https://github.com/netmod-wg/yang-next/issues/16)

Allow when in action

It would be good to be able to define an action when the value of a certain leaf is set to a (set of) specific value(s). For example: in case we want to reset specific HW components one might think to define a reset action, however certain components (identified by a HW class) might not be supporting a reset.

* *Github Labels*: complexity-low, backcompat-high, importance-med

* *Status*: open

Issue 17 (https://github.com/netmod-wg/yang-next/issues/17)

replace 'encoding' with 'representation'?

Discussion starts here: Lada says: > Yes, "representation" should be one of the terms newly defined in 7950bis, and it can also be mentioned that "encoding" was used informally in the past.

* *Github Labels*: complexity-low, backcompat-high, importance-low

* *Status*: open

Issue 18 (https://github.com/netmod-wg/yang-next/issues/18)

add a templating mechanism?

Templates are not a datastore-specific concept, they can equally well be used in artifacts (static documents)...

* *Github Labels*: complexity-high, backcompat-high, importance-med

* *Status*: open

Issue 19 (https://github.com/netmod-wg/yang-next/issues/19)

yang canonical integer format

The canonical format for integer types is the decimal representation. We do not seem to have a mechanism (other than a description statement) to declare that the canonical representation is lets say hexadecimal.

* *Github Labels*: backcompat-high, importance-low, complexity-unknown

* *Status*: open

Issue 20 (https://github.com/netmod-wg/yang-next/issues/20)

add explicit module version-stmt

Proposed a module-version statement to avoid parsing revision history for the current version. Considered redundant and addressed by versioning work (issue #45), so closed as duplicate.

* *Github Labels*: Duplicate

* *Status*: closed

Issue 21 (https://github.com/netmod-wg/yang-next/issues/21)

Restrict regex to a subset of XML regex specification

The choice in YANG to use XML regex has effectively meant that there is only a single library implementation of regex parsing. In most cases, the pattern statements only use the basic subset of the XML regex, and normally these pattern statements would validate against most standard regex engines with only a minimal amount of changes (it might be necessary to add anchors at the start and end of the line.

* *Github Labels*: complexity-high, backcompat-low, importance-low

* *Status*: open

Issue 22 (https://github.com/netmod-wg/yang-next/issues/22)

Restrict usage to a subset of XPATH

XPATH expressions can be complicated, hard to read, and it is easy to express stuff that (i) implementations don’t generally support, or (ii) are not what the author intended. I think that it would be useful to define a subset of XPATH that is allowed/valid, or possibly even consider defining a simpler YANG specific DSL.

* *Github Labels*: complexity-med, backcompat-med, importance-low

* *Status*: open

Issue 23 (https://github.com/netmod-wg/yang-next/issues/23)

add ’conformance-type’ leaf to ’import’ statement

Suggested indicating why a module is imported to aid implementers and catalogs. Comments questioned the problem being solved since tools can already infer usage. Closed.

* *Github Labels*: importance-low, backcompat-unknown, complexity-unknown

* *Status*: closed

Issue 24 (https://github.com/netmod-wg/yang-next/issues/24)

add a 'backwards-compatibility' leaf to 'revision' statement?

Proposed marking non-backwards-compatible changes in revision history to aid clients. Considered part of versioning work (issue #45) and closed as a duplicate.

* *Github Labels*: Duplicate

* *Status*: closed

Issue 25 (https://github.com/netmod-wg/yang-next/issues/25)

make yang more object-oriented

Suggested an object-oriented modeling approach where configuration resembles methods. Commenters felt this would be a different language and referenced limited traction in prior work. Closed.

* *Github Labels*: complexity-high, backcompat-low, importance-low, No support

* *Status*: closed

Issue 26 (https://github.com/netmod-wg/yang-next/issues/26)

Consider removing support for sub modules from YANG

Thread on NETMOD is 'Query about augmenting module from submodule in YANG 1.0'. Juergen: In case YANG 2.0 is ever done, I suggest someone files a proposal to remove submodules if the cost/benefit ratio is at odds.

* *Github Labels*: complexity-low, backcompat-low, importance-low

* *Status*: open

Issue 27 (https://github.com/netmod-wg/yang-next/issues/27)

Clarify YANG "status" keyword usage (e.g., hierarchical)

This issue requests: Clarify YANG "status" keyword usage (e.g., hierarchical).

* *Github Labels*: complexity-low, backcompat-high, importance-high

* *Status*: open

Issue 28 (https://github.com/netmod-wg/yang-next/issues/28)

add ’status’ as a sub statement to ’module’

So that an entire module (routing-cfg) can be deprecated w/o having to deprecate every node.

* *Github Labels*: complexity-low, backcompat-high, importance-med

* *Status*: open

Issue 29 (https://github.com/netmod-wg/yang-next/issues/29)

Clarify YANG validation for data nodes ’decoded’ from anydata to a tree of real nodes

Raised how offline validation tools should interpret anydata/anyxml in RPC inputs or configs and suggested using schema mount. Replies said this is a tooling concern and schema mount is not appropriate. Closed.

* *Github Labels*: complexity-low, importance-low, Not a YANG Issue, Tooling issue

* *Status*: closed

Issue 30 (https://github.com/netmod-wg/yang-next/issues/30)

key-predicate-expr should be (quoted-string / integer-value / decimal-value)

Proposed errata to allow numeric literals in instance-identifier key predicates instead of requiring quoted strings. The errata was withdrawn due to implementation impact, deferring the topic to YANG 2.0. Closed.

* *Github Labels*: complexity-low, backcompat-low, importance-low, Not an issue after all

* *Status*: closed

Issue 31 (https://github.com/netmod-wg/yang-next/issues/31)

Add 'clone' statement

Suggested a clone statement to copy existing schema nodes instead of defining reusable groupings. Concerns included misuse, clone loops, and increased complexity; the proposal was withdrawn. Closed.

* *Github Labels*: complexity-high, importance-low, No support

* *Status*: closed

Issue 32 (https://github.com/netmod-wg/yang-next/issues/32)

Allow Augmentation to Groupings

Proposed augmenting groupings directly so additions apply to all uses, avoiding repeated augment statements. Implementation and safety concerns dominated the discussion. Closed with no support.

* *Github Labels*: complexity-high, backcompat-low, No support

* *Status*: closed

Issue 33 (https://github.com/netmod-wg/yang-next/issues/33)

Tag YANG identity as an intermediate base for classification only

Some identity definitions are not intended to be actual values, but rather used as intermediate base definitions. YANG automation tools need a way to identify these nodes in a machine-readable manner.

* *Github Labels*: complexity-low, backcompat-high, importance-high

* *Status*: open

Issue 34 (https://github.com/netmod-wg/yang-next/issues/34)

Add native support for float/double

Add support for IEEE 754 binary floats and doubles. routing-types.yang (RFC 8294) defines bandwidth-ieee-float32, but ends up using a string as the base type (fine for text based encoding schemes, not so great for binary encoding schemes).

* *Github Labels*: complexity-med, backcompat-high, importance-high

* *Status*: open

Issue 35 (https://github.com/netmod-wg/yang-next/issues/35)

Simplify ’when’ statement processing

Proposed removing automatic deletion of nodes when when conditions become false due to complexity and NACM concerns. Discussion favored keeping current behavior and noted that offline validation does not involve request processing. Closed.

* *Github Labels*: Undesirable outcome

* *Status*: closed

Issue 36 (https://github.com/netmod-wg/yang-next/issues/36)

enable leafrefs to uniquely reference a nested list

Given the following data model: and a desire to uniquely reference a specific "bar": ‘ It’s not possible unless all the "bar" names are globally unique (not just for the current "foo").

* *Github Labels*: backcompat-high, importance-high, complexity-unknown

* *Status*: open

Issue 37 (https://github.com/netmod-wg/yang-next/issues/37)

YANG ANBF could be defined in a simpler way

Suggested a simplified ABNF that separates generic statement structure from argument validation for parser convenience. Responses noted the simplified grammar already exists in section 6.3 and that changing ABNF would be risky for low benefit. Closed.

* *Github Labels*: backcompat-high, importance-low, Undesirable outcome

* *Status*: closed

Issue 38 (https://github.com/netmod-wg/yang-next/issues/38)

Allow action to be invoked in the context of configuration datastore

Requested allowing actions on configuration datastores to support batch edits or complex operations. The discussion debated semantics and usefulness versus existing edit-config capabilities. Closed without adopting the change.

* *Github Labels*: Undesirable outcome

* *Status*: closed

Issue 39 (https://github.com/netmod-wg/yang-next/issues/39)

Allow deviation for description

Deviations can modify the meaning and content of a data node. However it is not possible to update relevant the description statement.

* *Github Labels*: complexity-low, backcompat-high, importance-high

* *Status*: open

Issue 40 (https://github.com/netmod-wg/yang-next/issues/40)

Allow deviation for Identities

It should be possible to declare that I do not support specific identities in a module. E.g.

* *Github Labels*: backcompat-high, importance-high, complexity-unknown

* *Status*: open

Issue 41 (https://github.com/netmod-wg/yang-next/issues/41)

Allow some references to from config-true to config-false (add capabilities)

IN some cases it would be needed to allow referencing config=false data from config=true leafrefs. E.g.

* *Github Labels*: complexity-high, importance-med, backcompat-unknown

* *Status*: open

Issue 42 (https://github.com/netmod-wg/yang-next/issues/42)

make schema-mount extension into a built-in statement

Proposed integrating schema-mount as a core language statement. Comments advised gaining more experience and using critical extensions if needed. Closed.

* *Github Labels*: complexity-med, backcompat-low, importance-low, Workaround is better

* *Status*: closed

Issue 43 (https://github.com/netmod-wg/yang-next/issues/43)

Support for multiple keys in a list

Asked for multiple key statements on a list to allow alternate unique access paths. Discussion noted optional keys were rejected and recommended using two lists with a shared grouping, despite drawbacks for config and augment. Closed.

* *Github Labels*: Not a YANG Issue, NETCONF issue

* *Status*: closed

Issue 44 (https://github.com/netmod-wg/yang-next/issues/44)

Clarify if multiple deviations target the same schema parts

This issue tracks the request titled ’Clarify if multiple deviations target the same schema parts’.

* *Github Labels*: complexity-low, backcompat-high, importance-high

* *Status*: open

Issue 45 (https://github.com/netmod-wg/yang-next/issues/45)

Refine YANG versioning

This issue tracks the request titled ’Refine YANG versioning’.

* *Github Labels*: backcompat-low, importance-high, complexity-unknown

* *Status*: open

Issue 46 (https://github.com/netmod-wg/yang-next/issues/46)

Binary encoding support (lets some types have binary persistence)

The YANG to CBOR encoding algorithms use binary types for some YANG data types. For example, enumerations are sent using the value number and bits are sent as a number with bits set corresponding to the position number.

* *Github Labels*: importance-high, backcompat-unknown, complexity-unknown, Not a YANG Issue, Workaround is better, NETMOD issue, NETCONF issue

* *Status*: open

Issue 47 (https://github.com/netmod-wg/yang-next/issues/47)

support external module requirements

Suggested requires/provides statements to express dependencies beyond import (e.g., augment relationships, common type modules). Reviewers indicated this belongs in YANG library or package definitions. Closed.

* *Github Labels*: Not a YANG Issue, NETMOD issue, NETCONF issue

* *Status*: closed

Issue 48 (https://github.com/netmod-wg/yang-next/issues/48)

when using a grouping, the designer should be able to modify just about any aspect of the grouping

Proposed broad modification capabilities for groupings (removing nodes, disabling if-feature) instead of copy/paste. Concerns focused on complexity and layering effects. Closed as an undesirable outcome.

* *Github Labels*: importance-low, backcompat-unknown, complexity-unknown, Undesirable outcome

* *Status*: closed

Issue 49 (https://github.com/netmod-wg/yang-next/issues/49)

Introduce critical extensions

Critical extensions would be allowed to extend or modify YANG semantics and would be mandatory to implement. In fact, some existing extensions, such as *yang-data*, are already of this category, even though RFC 7950 only permits extensions to address _non-YANG semantics_.

* *Github Labels*: complexity-high, importance-med, backcompat-unknown

* *Status*: open

Issue 50 (https://github.com/netmod-wg/yang-next/issues/50)

YANG Packages (multi-module conformance/guidance)

Requested a machine-readable way to define packages of modules with version/feature requirements. Discussion suggested this is better handled by YANG library or instance data documents and pursued in a separate draft, not the YANG language itself. Closed as not a YANG issue.

* *Github Labels*: Not a YANG Issue, NETMOD issue

* *Status*: closed

Issue 51 (https://github.com/netmod-wg/yang-next/issues/51)

A general way to add stmts to any part of a schema

As a vendor we sometimes want to extend a YANG model by annotating it with extra metadata information. E.g.

* *Github Labels*: backcompat-high, importance-low, complexity-unknown

* *Status*: open

Issue 52 (https://github.com/netmod-wg/yang-next/issues/52)

add a statement for modeling checkbox dialogs

Proposed a statement to model multi-select checkbox dialogs, similar to choice for radio buttons. Alternatives included using optional leafs, bits, or UI annotations/overlays. Closed with preference for the overlay/annotation solution.

* *Github Labels*: complexity-low, backcompat-high, importance-low, Workaround is better

* *Status*: closed

Issue 53 (https://github.com/netmod-wg/yang-next/issues/53)

Create a way for a statement to tie-in with augment/deviation

RFC7950 section 7.19 specifically says: ‘ An extension can allow refinement (see Section 7.13.2) and deviations (Section 7.20.3.2), but the mechanism for how this is defined is outside the scope of this specification.‘ Via refine this extends to augment and uses.

* *Github Labels*: complexity-med, importance-low, backcompat-unknown, Need Examples / Use Case

* *Status*: open

Issue 54 (https://github.com/netmod-wg/yang-next/issues/54)

Define a way for extensions to declare sub-statement validity

RFC7950 leaves this capability out in section 7.19: ‘ The substatements of an extension are defined by the "extension" statement, using some mechanism outside the scope of this specification. Syntactically, the substatements MUST be YANG statements, including extensions defined using "extension" statements.

* *Github Labels*: complexity-med, backcompat-high, importance-med

* *Status*: open

Issue 55 (https://github.com/netmod-wg/yang-next/issues/55)

define an encoding-independent "ypath:1.0" type

Discussed the need for a context-independent encoding of XPath expressions and YANG types, avoiding prefix-to-namespace reliance. The conclusion was that this can be addressed in RFC6991bis by making XPath 1.0 context independent, without changing YANG. Closed.

* *Github Labels*: Workaround is better, Undesirable outcome

* *Status*: closed

Issue 56 (https://github.com/netmod-wg/yang-next/issues/56)

context-independent encoding of instance-identifiers and identityrefs

In the XML encoding of an instance-identifier or an identityref, there are prefixes that are supposed to be defined in the XML document. This means that there is no canonical format of these values.

* *Github Labels*: complexity-med, backcompat-low, importance-high

* *Status*: open

Issue 57 (https://github.com/netmod-wg/yang-next/issues/57)

introduce if-module

it would be useful to have a statement "if-module" that can be used like if-feature: leaf my-interface { if-module ietf-interfaces; type if:interface-ref; }.

* *Github Labels*: complexity-low, backcompat-high, importance-unknown, Need Examples / Use Case

* *Status*: open

Issue 58 (https://github.com/netmod-wg/yang-next/issues/58)

Introduce XPath function datastore()

The expression datastore(name) would select the root node of datastore name, where name is the name of a datastore. YANG rules for accessible trees are sometimes too limiting.

* *Github Labels*: complexity-high, backcompat-high, importance-low

* *Status*: open

Issue 59 (https://github.com/netmod-wg/yang-next/issues/59)

Preliminary Status

Sometimes we deliver features to customers before they are stable, to give them a working preview of functionality. For this we use the following extension statement extension preliminary { description "The schema node is part of an early design effort.

* *Github Labels*: complexity-low, backcompat-high, importance-med

* *Status*: open

Issue 60 (https://github.com/netmod-wg/yang-next/issues/60)

Allowing module private groupings, typedefs

Within a module. top level groupings and typedefs are externally visible and reusable by other modules, whereas local groupings and typedefs are private to the module, and can be changed in a BC way.

* *Github Labels*: complexity-low, backcompat-high, importance-med

* *Status*: open

Issue 61 (https://github.com/netmod-wg/yang-next/issues/61)

Make NACM default-deny-all and default-deny-write built-in statements

Ready to promote these to language statements so it is not NACM-specific.

* *Github Labels*: complexity-low, backcompat-high, importance-med

* *Status*: open

Issue 62 (https://github.com/netmod-wg/yang-next/issues/62)

Currently, list keys are all mandatory - allows default values for keys.

This issue tracks the request titled ’Currently, list keys are all mandatory - allows default values for keys.’.

* *Github Labels*: backcompat-high, importance-med, complexity-unknown

* *Status*: open

Issue 63 (https://github.com/netmod-wg/yang-next/issues/63)

Should it be possible to deviate "status"?

Should it be possible for an implementation to modify the status of a datanode? E.g.

* *Github Labels*: complexity-med, backcompat-high, importance-med

* *Status*: open

Issue 64 (https://github.com/netmod-wg/yang-next/issues/64)

Clarify if double quotes are needed around identifiers

Questioned whether identifiers need quotes when used as string arguments (e.g., defaults). Discussion concluded that quoting is optional when no whitespace or special characters are present and that the grammar should remain permissive; pretty printers may normalize. Closed as not a YANG issue.

* *Github Labels*: Not a YANG Issue, NETMOD issue

* *Status*: closed

Issue 65 (https://github.com/netmod-wg/yang-next/issues/65)

Require implementation of "status deprecated" data nodes

We should change YANG (with some sensible migration path) to require servers to implement status deprecated data nodes or otherwise use explicit deviations to indicate that the data nodes are not implemented.

* *Github Labels*: complexity-low, backcompat-high, importance-high

* *Status*: open

Issue 66 (https://github.com/netmod-wg/yang-next/issues/66)

Require that "status obsolete" nodes are not implemented

We should consider changing YANG (with some sensible migration path) to require servers to not implement "status obsolete" nodes or otherwise use a explicit deviation mechanism to indicate that the data noes are still implemented and present in the schema.

* *Github Labels*: complexity-low, backcompat-low, importance-med

* *Status*: open

Issue 67 (https://github.com/netmod-wg/yang-next/issues/67)

clarify "require-instance" property for leafrefs

Asked when require-instance is evaluated and how errors are reported for delete operations of referenced nodes. Responses pointed to existing text (sections 8.3.3 and 15.5) covering these cases. Closed with no change required.

* *Github Labels*: Not an issue after all

* *Status*: closed

Issue 68 (https://github.com/netmod-wg/yang-next/issues/68)

Add if-feature on must stmt (removed "on import stmt" part)

When designing the IGMP Proxy model, I feel it needs adding if-feature in import stmt and must stmt. But it doesn’t support now.

* *Github Labels*: complexity-med, backcompat-high, importance-low

* *Status*: open

Issue 69 (https://github.com/netmod-wg/yang-next/issues/69)

Clarify ’deviation’ substatements to match ABNF grammar

As mentioned in: The current table in RFC7950, Section 7.20.3.2 lists all permitted substatements however each argument to "delete" is treated separately per ABNF. If we are not going to add other possible substatements (e.g.

* *Github Labels*: complexity-low, backcompat-high, importance-high

* *Status*: open

Issue 70 (https://github.com/netmod-wg/yang-next/issues/70)

Introduce support for critical annotations

Similar to #49, but for annotations (RFC 7952), so that something like Juniper’s "inactive" annotation can be introduced without requiring a new version of YANG.

* *Github Labels*: backcompat-unknown, complexity-unknown, importance-unknown, Need Examples / Use Case

* *Status*: open

Issue 71 (https://github.com/netmod-wg/yang-next/issues/71)

extension-stmt conformance

The conformance requirements and capabilities discovery for extension statements is broken and implementations do not follow it. If a module contains extensions and a server advertises the implemented module in the YANG library, then the server must implement all extensions in the module.

* *Github Labels*: backcompat-unknown, complexity-unknown, importance-unknown, Need Examples / Use Case

* *Status*: open

Issue 72 (https://github.com/netmod-wg/yang-next/issues/72)

Introduce an annotation that resolves the union member

Determining the union member to be used for a given instance is a fragile and potentially demanding process that may also depend on the instance representation. The (optional) annotation would pinpoint the member type so that no guesswork is needed.

* *Github Labels*: complexity-med, importance-high, backcompat-unknown

* *Status*: open

Issue 73 (https://github.com/netmod-wg/yang-next/issues/73)

Initial value

3GPP is starting to use YANG for modeling. The "default value" as defined by 3GPP is actually an initial value.

* *Github Labels*: complexity-med, importance-med

* *Status*: open

Issue 74 (https://github.com/netmod-wg/yang-next/issues/74)

Support for unique leaf (or leaf combo) in a list of lists

If I have a list which has a container, I can specify a leaf to be unique or a combination of leaf nodes to be unique. e.g.: If I have a list of lists, I know of no way to have a leaf inside the lower list to be delcared unique across all lists.

* *Github Labels*: complexity-med, importance-low

* *Status*: open

Issue 75 (https://github.com/netmod-wg/yang-next/issues/75)

Deprecate "import by exact revision"

Using YANG import with an exact revision date is invariably the wrong thing to do and I propose that it should be removed from the next version of the language (but allowed for modules using yang-version 1.1).

* *Github Labels*: complexity-low, importance-high

* *Status*: open

Issue 76 (https://github.com/netmod-wg/yang-next/issues/76)

Make submodule "include" statement require a revision date.

A particular module revision should be a precisely defined immutable instance of a YANG module. However, allowing submodule includes to not specify the revision of the submodules that they include allows this to be broken.

* *Github Labels*: importance-med, complexity-unknown

* *Status*: open

Issue 77 (https://github.com/netmod-wg/yang-next/issues/77)

Add support for a tuple type

Sometimes it is desirable to group multiple leaves together into the same value type that is available on a single path. Adding native support for something like a tuple type might be an elegant way of solving this, rather than the current approach of using an adhoc string based tuple type.

* *Github Labels*: complexity-high, importance-high

* *Status*: open

Issue 78 (https://github.com/netmod-wg/yang-next/issues/78)

allow 'must' as a sub statement to 'grouping'

The crypto-types:asymmetric-key-pair-grouping grouping is a "container-less" grouping (a grouping that defines its descendents directly, without wrapping them with a container), which makes it impossible to define a 'must' statement that spans all its descendents (none of which are "mandatory true").

* *Github Labels*: importance-low

* *Status*: open

Issue 79 (https://github.com/netmod-wg/yang-next/issues/79)

add 'unique' as a substatement to 'leaf-list'

The request sought a unique substatement for leaf-lists. The discussion noted RFC7950 already requires uniqueness for configuration leaf-lists, and the issue was closed as not applicable.

* *Github Labels*: Not an issue after all

* *Status*: closed

Issue 80 (https://github.com/netmod-wg/yang-next/issues/80)

enable a server express conformance to a set of identifiers

This issue requests: enable a server express conformance to a set of identifiers.

* *Github Labels*: importance-high

* *Status*: open

Issue 81 (https://github.com/netmod-wg/yang-next/issues/81)

let 'description' be a substatement to 'input' and 'output'

It would be nice to have a description statement to describe what a collection of input/output descendent nodes are for.

* *Github Labels*: complexity-low, importance-med

* *Status*: open

Issue 82 (https://github.com/netmod-wg/yang-next/issues/82)

enable features to be supported per grouping-use (not globally per-datastore)

I’m unsure if this is a YANG Next, YANG library, or a modeling issue... Use case: a shared low-level module defines a feature and defines a grouping with an if-feature with it.

* *Github Labels*: complexity-high, importance-low

* *Status*: open

Issue 83 (https://github.com/netmod-wg/yang-next/issues/83)

Clarify canonical representation of typedefs

Some typedefs (and possibly some leaves) (e.g. ipv6-prefix) define their own canonical representation of the value space that overrides the canonical representation defined in the YANG built in types (RFC 7950 section 9.1).

* *Github Labels*: importance-high, complexity-unknown

* *Status*: open

Issue 84 (https://github.com/netmod-wg/yang-next/issues/84)

allow notifications/actions to appear in invalid contexts

For example, RFC 7950 Section 7.16 says: But sometimes a grouping contains a notification or an action, and thus is currently disqualified from being _used_ inside (in this case) a notification.

* *Github Labels*: complexity-low, backcompat-high, importance-high

* *Status*: open

Issue 85 (https://github.com/netmod-wg/yang-next/issues/85)

add "uses" as a sub-statement to "augment"

Asked to allow reuse of identical augment blocks via a uses substatement. Closed because this is already supported in YANG 1.1.

* *Github Labels*: none

* *Status*: closed

Issue 86 (https://github.com/netmod-wg/yang-next/issues/86)

allow ’case’ as a substatement to ’grouping’

Sometimes there is a need to augment the same case statement into multiple choice statements. Ideally the case statement itself could also be defined in the grouping statement.

* *Github Labels*: importance-low

* *Status*: open

Issue 87 (https://github.com/netmod-wg/yang-next/issues/87)

enable specifying augmentation’s location amongst siblings

Requested control over the placement of augmented nodes in resolved trees or diagram output. Comments noted YANG is unordered and enforcing order would be a large change; tooling could handle presentation. Closed.

* *Github Labels*: none

* *Status*: closed

Issue 88 (https://github.com/netmod-wg/yang-next/issues/88)

clarify NP-containers

Searching the NETCONF list archives for the word "non-presense" found the following: * 2019-06-21: restconf ’get’ on non-presence container * 2018-06-28: Existence of Non-Presence Containers * 2016-07-11: What should a server response be?

* *Github Labels*: complexity-low, importance-med

* *Status*: open

Issue 89 (https://github.com/netmod-wg/yang-next/issues/89)

add ’recommended-default’ statement

It is important to maintain backward compatibility with existing program behavior. Often the YANG default is not the recommended setting, but rather the setting that matches current program behavior.

* *Github Labels*: importance-low

* *Status*: open

Issue 90 (https://github.com/netmod-wg/yang-next/issues/90)

Have a mechanism to allow enums to be extended

I think that in some cases, individuals writing YANG modules end up using identities instead of enums because they want to leave the door open to them being extended in future.

* *Github Labels*: importance-high, complexity-unknown

* *Status*: open

Issue 91 (https://github.com/netmod-wg/yang-next/issues/91)

Consider relaxing identity uniqueness to only require uniqueness with the base identity hierarchy

Flagging this only because it came up as a discussion point on yang-doctors. Currently identities must be unique within a module. So, in the case of loopback, this meant that I originally defined "loopback-internal", "loopback-line", "loopback-connector" as descendants of base identity "loopback" to avoid namespace collisions.

* *Github Labels*: complexity-high, importance-low

* *Status*: open

Issue 92 (https://github.com/netmod-wg/yang-next/issues/92)

enable if-feature statements to be "refined" into notifications and actions

A grouping can define a data tree, as well as notifications, actions. The refine statement can add if-feature statements to many types of data nodes, but not for actions or notifications.

* *Github Labels*: complexity-low, importance-high

* *Status*: open

Issue 93 (https://github.com/netmod-wg/yang-next/issues/93)

support 'dynamic default'

if a leaf node can not be defined a static default value, but it has a uncertain default value when it is created. For example: If list 'foo' is created, if leaf named 'type''s value is foo1, the default value of leaf named 'value' is 10, if leaf named 'type''s value is foo2, the default value of leaf named 'value' is 20.

* *Github Labels*: complexity-high, importance-low

* *Status*: open

Issue 94 (https://github.com/netmod-wg/yang-next/issues/94)

deref() function for leafref statements

This issue requests: deref() function for leafref statements.

* *Github Labels*: complexity-low, importance-high

* *Status*: open

Issue 95 (https://github.com/netmod-wg/yang-next/issues/95)

Allow an module import to be defined as "types only"

Today, when a module imports another module it isn't made explicit whether that import is to fulfill a path or identity dependency or simply for a typedef dependency. Whilst working on YANG packages, I've began to think that it might be helpful to allow import statements to be annotated to indicate that it only imports type definitions, making the relationship between YANG modules a bit more clear.

* *Github Labels*: importance-high

* *Status*: open

Issue 96 (https://github.com/netmod-wg/yang-next/issues/96)

Clarify definitions of YANG schema tree, vs module schema tree

The following errata was proposed (but likely rejected) for RFC 7950: Original Text ------------- o schema tree: The definition hierarchy specified within a module. Corrected Text -------------- o schema tree: The hierarchy of schema nodes defined in the set of all modules implemented by a server, as specified in the YANG library data .

* *Github Labels*: complexity-low, backcompat-high, importance-med

* *Status*: open

Issue 97 (https://github.com/netmod-wg/yang-next/issues/97)

enable 'ordered-by' to be refined into a list or leaf-list

A grouping may define a list or leaf-list and, for whatever reason, the using model wished to change the 'ordered-by' value. This should be allowed, right?

* *Github Labels*: none

* *Status*: open

Issue 98 (https://github.com/netmod-wg/yang-next/issues/98)

Disallow if-feature statements where enabling the feature removes data nodes from the schema

The expressions for if-feature nodes should not be allowed to remove nodes from the tree when a feature is enabled because it makes conformance for clients very complex. E.g.

* *Github Labels*: backcompat-low, importance-unknown

* *Status*: open

Issue 99 (https://github.com/netmod-wg/yang-next/issues/99)

Clarify duplicate revision dates used in revision history

The RFC does not mention any error condition if multiple revision-stmts have the same revision-date value. This occurs in openconfig-inet-types.yang and other real modules.

* *Github Labels*: importance-med

* *Status*: open

Issue 100 (https://github.com/netmod-wg/yang-next/issues/100)

Add errata-stmt to YANG

There needs to be a way to specify the known Errata corrections to a YANG module. There are many published modules with bugs.

* *Github Labels*: importance-high

* *Status*: open

Issue 101 (https://github.com/netmod-wg/yang-next/issues/101)

allow 'require-instance' to be refined

It would be nice to leafref (inside a grouping)'s require-instance property. This to compliment when the list the leafref refers to was itself refined from config true to config false.

* *Github Labels*: complexity-low, backcompat-high, importance-med

* *Status*: open

Issue 102 (https://github.com/netmod-wg/yang-next/issues/102)

ascii vs. unicode strings

Per Alexey Melnikov's DISCUSS on the module-tags draft, though the issue is a long-standing one...

* *Github Labels*: importance-low

* *Status*: open

Issue 103 (https://github.com/netmod-wg/yang-next/issues/103)

Clarify implicit 'case' behavior

Implicit case statements have some interesting undocumented behavior:.

* *Github Labels*: complexity-low, backcompat-high, importance-med

* *Status*: open

Issue 104 (https://github.com/netmod-wg/yang-next/issues/104)

clarify "instance-required" behavior in typedefs

As discussed in authors expect that the require-instance statement is available not only for leafref and instance-identifier types, but also for all the types derived from them using typedef statement.

* *Github Labels*: complexity-low, backcompat-high, importance-high

* *Status*: open

Issue 105 (https://github.com/netmod-wg/yang-next/issues/105)

remove the "anyxml" statement

Synopsis: The "anyxml" statement was made in RFC 6020, before JSON support was added, but its value is questionable now... Propose to remove (if YANG-next == YANG 2.0) or deprecate (if YANG-next == YANG 1.2).

* *Github Labels*: complexity-low, importance-med, backcompat-unknown

* *Status*: open

Issue 106 (https://github.com/netmod-wg/yang-next/issues/106)

Allow "input/output" to be defined without any child data nodes

Change the ABNF to allow an input and output statements to be defined without any data nodes (e.g. in the case that models are expected to augment the input/output statement).

* *Github Labels*: complexity-low, backcompat-high, importance-med

* *Status*: open

Issue 107 (https://github.com/netmod-wg/yang-next/issues/107)

Add deviate(not-supported) support for identities

There is no way for a server vendor to tell the YANG client that a specific identity in a YANG module is not supported. This is critical for vendors who use YANG modules defined by an SDO (all of them) YANG conformance for identities is particularly awful at this time because it mandates that every identity in a supported module be implemented.

* *Github Labels*: complexity-med, importance-high

* *Status*: open

Issue 108 (https://github.com/netmod-wg/yang-next/issues/108)

Allow when in notification

It seems the when statement should be supported also for the notification statement with the exact same justification as for action (.

* *Github Labels*: importance-med, backcompat-unknown, complexity-unknown

* *Status*: open

Issue 109 (https://github.com/netmod-wg/yang-next/issues/109)

Change and compatibility rules for config=false (state) data

Allowed changes and what is compatible should be specified differently for config=true and config=false data. E.g.

* *Github Labels*: complexity-high, importance-med

* *Status*: open

Issue 110 (https://github.com/netmod-wg/yang-next/issues/110)

Clarify canonical order in RFC 7950

RFC 7950 states this: The ABNF grammar defines the canonical order. But the ABNF also has sections that state: ;; these stmts can appear in any order These two statements seem to cause some confusion, and hence it might be helpful to clarify that the ";; these stmts can appear in any order" does not remove the need to list statements in the order listed in the ABNF for them to be regarded as being in canonical order.

* *Github Labels*: complexity-low, backcompat-high, importance-med

* *Status*: open

Issue 111 (https://github.com/netmod-wg/yang-next/issues/111)

Clarify whether revision-dates must be unique or not

RFC 7950 does not explicitly state whether two different revisions of a module can be published with the same date. There is an example of an OpenConfig YANG module that has published multiple revisions with the same revision date, and the OpenConfig proponents state that RFC 7950 does not disallow this.

* *Github Labels*: complexity-low, backcompat-high, importance-high

* *Status*: open

Issue 112 (https://github.com/netmod-wg/yang-next/issues/112)

Support for conditional default values

This issue tracks the request titled 'Support for conditional default values'.

* *Github Labels*: importance-low

* *Status*: open

Issue 113 (https://github.com/netmod-wg/yang-next/issues/113)

Treat if-feature which references a non-existing feature as valid YANG but not enabled

Scenario: A software product comes out with a YANG model. It supports extensions referencing its model.

* *Github Labels*: importance-med

* *Status*: open

Issue 114 (https://github.com/netmod-wg/yang-next/issues/114)

Don't treat non-exist import module as a error if there is no any effective reference to this import module in current module

Suggested not requiring imported modules when deviations remove all effective references. Reviewers said this is not implementable because deviations are separate modules and prefixes used at parse time must resolve. Closed as non-implementable.

* *Github Labels*: none

* *Status*: closed

Issue 115 (https://github.com/netmod-wg/yang-next/issues/115)

Clarify the meaning of properties which have default value

Some statements have default value, such as config/mandatory/min-elements/max-elements/status etc. if these statements are not occur under parent nodes, and should we think the parent nodes have these properties actually?

* *Github Labels*: complexity-med, importance-high

* *Status*: open

Issue 116 (https://github.com/netmod-wg/yang-next/issues/116)

Clarify the behavior if the schema nodes in XPath expression are un-supported

If the schema nodes in XPath expression are un-supported by deviation, should the YANG parser report error? If YANG parser report error, user have to also change the XPath expression by deviation (if its when, it can not be deviated.) If YANG parser dont report error, the evaluation of XPath expression MAY cause unexpected result.

* *Github Labels*: complexity-med, backcompat-high, importance-med

* *Status*: open

Issue 117 (https://github.com/netmod-wg/yang-next/issues/117)

Add dynamic feature to YANG

YANG provide the capability to define data model statically. But it can not meet the requirements of some scenarios: Some nodes are associated with licenses.

* *Github Labels*: importance-low

* *Status*: open

Issue 118 (https://github.com/netmod-wg/yang-next/issues/118)

Co-existence between YANG1.0 and YANG1.1

Argued that YANG 1.0 modules should be able to import YANG 1.1 modules and that RFC7950 has protocol-specific language. Responses emphasized that YANG 1.0 semantics must apply to all imports and that new keywords complicate this. Closed with consensus to keep current rules, though protocol wording could be generalized.

* *Github Labels*: none

* *Status*: closed

Issue 119 (https://github.com/netmod-wg/yang-next/issues/119)

The next step of XPath

Scenarios: The expression of when/must of YANG using XPath 1.0 is not readable and error-prone. The XPath filtering of NETCONF has the same problem.

* *Github Labels*: complexity-high, importance-med

* *Status*: open

Issue 120 (https://github.com/netmod-wg/yang-next/issues/120)

allow more restriction on Identity-ref

Proposed permit/deny substatements under identityref base to exclude specific derived identities. Discussion favored a simpler restriction mechanism and noted overlap with other issues. Closed as a duplicate.

* *Github Labels*: none

* *Status*: closed

Issue 121 (https://github.com/netmod-wg/yang-next/issues/121)

Add support for hierarchical default values

Hierarchical configuration is often expressed in YANG models. In this model, only the "default" statement should be placed at the root of the hierarchical configuration, and all other nodes in the configuration hierarchy must not have a default statement, and instead are required to state in text that the default value is inherited from the next node up in the hierarchical config chain.

* *Github Labels*: importance-med, Need Examples / Use Case

* *Status*: open

Issue 122 (https://github.com/netmod-wg/yang-next/issues/122)

Changing an identity base

Since, as explained in section 7.18.2 of RFC7950, the derivation of identities is transitive, replacing a "base" statement with new "base" statement which is derived from the previous one is also a BC change.

* *Github Labels*: importance-med, complexity-unknown

* *Status*: open

Issue 123 (https://github.com/netmod-wg/yang-next/issues/123)

Allow identities to be active even when module is not implemented

Currently a module has to be implemented in order for its identities to be active. Why is this so?

* *Github Labels*: importance-high

* *Status*: open

Issue 124 (https://github.com/netmod-wg/yang-next/issues/124)

Add ability to deviate or change status of an identity

E.g., for deprecated security protocols, a server should be able to indicate which ones it supports (although this could also be done via a capabilities YANG module).

* *Github Labels*: importance-high

* *Status*: open

Issue 125 (https://github.com/netmod-wg/yang-next/issues/125)

Further restrict YANG Unicode to exclude "del" and C0 control characters

See Unicode Assignables, section 4.3, that defines the table of Unicode assignable characters. This matches what is in RFC 7950, 'yang-char', page 209, except that YANG allows characters in the range %x20-D7FF, but draft-bray-unichars-07 splits this into two ranges to also exclude C1 control characters and DEL: It would

probably be helpful for the next version of YANG to be updated to also exclude these characters, and perhaps reference one of the "Character Repertoires", if that draft-bray-unichars progresses.

* *Github Labels*: complexity-low

* *Status*: open

Issue 126 (https://github.com/netmod-wg/yang-next/issues/126)

Injection of circular imports by deviation

RFC 7950 is unclear on whether import dependencies may be introduced by way of deviations. According to RFC 7950: This is clear and I believe most people find this reasonable enough.

* *Github Labels*: importance-high

* *Status*: open

Issue 127 (https://github.com/netmod-wg/yang-next/issues/127)

Relax rules for identityref representation without a prefix

Problem There are several YANG modules that have must/when XPath that compares an identityref leaf to a string literal. uses terminal-otn-protocol-top { when "config/logical-channel-type = 'PROT_OTN'" { description "Include the OTN protocol data only when the channel is using OTN framing."; } } The strict rules in the RFC (sec 9.10) say that the identity (e.g.

* *Github Labels*: complexity-low, backcompat-high, importance-med

* *Status*: open

Issue 128 (https://github.com/netmod-wg/yang-next/issues/128)

Relax rules on usage of deprecated or obsolete identifiers

sec 7.21.2 Issue 1) these MUST requirements make a YANG module brittle and hard to change Issue 2) "deprecated" status should be treated as MUST implement, so "current" using "deprecated" is just a warning Issue 3) "obsolete" status should be treated as MUST NOT implement, but it actually is SHOULD NOT implement.

* *Github Labels*: importance-med

* *Status*: open

Issue 129 (https://github.com/netmod-wg/yang-next/issues/129)

Enable reusability without groupings

The "grouping" statement is great, but only if the module-designer had the foresight to create groupings, which rarely happens. The YANG Full Embed draft attempts to address this by enabling other nodes in a module (e.g., leaf, container) to be used like a grouping.

* *Github Labels*: importance-high

* *Status*: open

Issue 130 (https://github.com/netmod-wg/yang-next/issues/130)

Add 'deprecated' statement

A new statement is needed to help enforce professional API lifecycle management. The 'deprecated' statement contains information about the associated object which has a status of 'deprecated' The purpose of the statement is to provide machine and human readable instructions about the deprecation and suggest a replacement description: text why the node is deprecated replaced-by: machine-readable identifier to use instead of this identifier TBD: expected timeframe for obsolete status container foo { status deprecated; deprecated { description "This node replaced by new system module"; replaced-by bar; } }.

* *Github Labels*: complexity-low, backcompat-high, importance-high

* *Status*: open

Issue 131 (https://github.com/netmod-wg/yang-next/issues/131)

make annotation a built-in YANG statement

RFC 7952 says: An NBC version of YANG-next can do it.

* *Github Labels*: complexity-low, importance-med

* *Status*: open

Issue 132 (https://github.com/netmod-wg/yang-next/issues/132)

Allow config=true list without a key-stmt

Suggested permitting keyless config lists for cases where the list is only created or replaced as a whole. Opponents cited interoperability and quality issues with keyless lists, while others noted autokey expectations. Closed as a duplicate of another issue.

* *Github Labels*: none

* *Status*: closed

Issue 133 (https://github.com/netmod-wg/yang-next/issues/133)

support for more efficient CBOR representation of strings

There are some strings that could be smaller to encode as binary than the UTF8 representation. Strings such as ipv4-address can be encoded more efficiently n binary YANG to CBOR encoding is being enhanced with some new work to encode some strings in CBOR binary format This should not be done with a hard-wired solution.

* *Github Labels*: importance-med

* *Status*: open

Issue 134 (https://github.com/netmod-wg/yang-next/issues/134)

Apply verified errata and resolve held for document update errata

There are 14 verified errata RFC 7950 Errata There are 2 held for review Held for Document Update Must Do.

* *Github Labels*: complexity-low, backcompat-high, importance-high

* *Status*: open

Issue 135 (https://github.com/netmod-wg/yang-next/issues/135)

Make YANG Semver real statements instead of extensions

YANG Semver It is important that "imports" and other procedures directly related to YANG language processing are mandatory to implement so all tools get the correct results. extensions should be real (mandatory-to-implement) statements and not optional extensions no intent to change Semver work file name issues are not be included in this issue Must Do.

* *Github Labels*: complexity-med, importance-med

* *Status*: open

Issue 136 (https://github.com/netmod-wg/yang-next/issues/136)

Revise Module Update Rules

module update rules Dreaded "Section 11" had caused more trouble than any other section. *NBC Change* New Identifier Not Mandatory WG list discussion has resulted in the following change text change in this section OLD NEW Other refinements may be identified during development Must Do: complexity: medium, bc: low, importance: high.

* *Github Labels*: complexity-med, importance-med

* *Status*: open

Issue 137 (https://github.com/netmod-wg/yang-next/issues/137)

Add validation-rules-stmt to identify YANG validation scope

Proposed a validation-rules statement with strict/relaxed/off modes to handle offline validation, multi-datastore references, or template-like configuration. Discussion questioned the examples and scope, suggesting use of intended datastores, must statements, or presence-like behavior. Closed per the Oct 24 meeting.

* *Github Labels*: none

* *Status*: closed

Issue 138 (https://github.com/netmod-wg/yang-next/issues/138)

Add new node type 'listref'

This is related to tuples #77 A listref is similar to a leafref but it represents the keys or position that identtify one list instance A listref is a terminal node like a leaf or a leaf-list or anydata A listref works the same no matter how many key leafs are defined for the list *Full Representation* no -keys: position of the list entry keys: leaf matching each leaf in the key-stmt *Short Representation* no-keys: use uint32 keys: use TBD tuple from #77 solution Example target grouping Pointed-at list: Listref A listref value example.

* *Github Labels*: importance-high

* *Status*: open

Issue 139 (https://github.com/netmod-wg/yang-next/issues/139)

Clarify adding mandatory nodes with augment + when

augments para 7 example is not valid since old client knows about values 1 .. 5 for toasterWeight.

* *Github Labels*: complexity-low, backcompat-high, importance-med

* *Status*: open

Issue 140 (https://github.com/netmod-wg/yang-next/issues/140)

grouping usage requirements

There are some design choices that are possible with YANG groupings that need to be considered for the uses-stmt There are documentation guidelines but these are not commonly followed, or complete.

* *Github Labels*: complexity-med, importance-med

* *Status*: open

Issue 141 (https://github.com/netmod-wg/yang-next/issues/141)

Add automatic list key generation (autokey)

There are many use-cases where the list key does not contain any semantics other than being a unique identifier. In this mode, the designer does not want to define or implement the list key management.

* *Github Labels*: importance-med

* *Status*: open

Issue 142 (https://github.com/netmod-wg/yang-next/issues/142)

Allow if-feature on deviations

The issue requested allowing "if-feature" under "deviation" so a single deviation module could cover both feature-enabled and feature-disabled cases. It was closed as a duplicate of issue #2.

* *Github Labels*: none

* *Status*: closed

Issue 143 (https://github.com/netmod-wg/yang-next/issues/143)

Allow deviation-stmt within uses-stmt

Sometimes an existing grouping is almost usable in a new us-case. YANG 1.1 allows some refinements in the 'uses' expansion, but no deviations.

* *Github Labels*: complexity-med, importance-med

* *Status*: open

Issue 144 (https://github.com/netmod-wg/yang-next/issues/144)

YANG 1.1 translation

A standard translation from YANG 2.0 to YANG 1.1 should be defined. It will be difficult to introduce YANG 2.0 modules if a YANG 1.1 version cannot be automatically generated for existing tools to use.

* *Github Labels*: complexity-low, importance-high

* *Status*: open

Issue 145 (https://github.com/netmod-wg/yang-next/issues/145)

Let a presence container have a default presence?

A presence container is like a leaf of type boolean. Leafs can have a default have, but not P-containers, why?

* *Github Labels*: importance-low

* *Status*: open

Issue 146 (https://github.com/netmod-wg/yang-next/issues/146)

Consistent default value behavior for operations and validation

YANG validation (e.g., must-stmt) sometimes depends on whether a "node exists in the accessible tree". This creates a cornercase with the default-stmt The issue is that every server can decide differently whether node /npcon/B is in the accessible tree.

* *Github Labels*: complexity-low, backcompat-high, importance-high

* *Status*: open

Issue 147 (https://github.com/netmod-wg/yang-next/issues/147)

New default-system statement to indicate that the default value is not constant (determined by the system)

See which we agreed is of low-importance. When the default is determined by the system (conditions usually mentioned in the description), it'd be handy for tooling to know that such is the case.

* *Github Labels*: complexity-low, backcompat-high, importance-med

* *Status*: open

Issue 148 (https://github.com/netmod-wg/yang-next/issues/148)

YANG++

YANG++ YANG++ is a data modeling language under development. Project Goals Complete the design and specification of the YANG++ language Design and implement open-source plugins - Native compiler support for YANG++ - YANG 1.1 Translation Tool Documentation YANG++ DRAFT.

* *Github Labels*: importance-high

* *Status*: open

Issue 149 (https://github.com/netmod-wg/yang-next/issues/149)

Error statement for actions rpcs

In some cases for rpcs and actions there is a need to supply detailed error information beyond what is specified in RFC 6241. NETCONF allows for additional information in the "error-message" or "error-info" elements, however there is no way to define the structure and syntax of the error message.

* *Github Labels*: complexity-low, importance-high

* *Status*: open

Issue 150 (https://github.com/netmod-wg/yang-next/issues/150)

nbc-change-stmt

There is a need for a new statement to identify an NBC change in an exportable statement. The Semver extension is not helpful since it is at the module-level instead of the object level, This cannot be

done with the ’deprecated’ statement in #130 since the NBC change is most likely done in a ’current’ definition with no intent to deprecate and replace it.

* *Github Labels*: none

* *Status*: open

Issue 151 (https://github.com/netmod-wg/yang-next/issues/151)

YANG profiles and views

There a trend (especially in IETF WGs) to create new YANG models to address some specific use cases rather than re-using existing YANG models because existing YANG models either provides more information than needed for that specific application or the information provided is not formatted as expected by the application Slides presented and discussed during the IETF 121 side meeting: YANG Profiles and Views v00.pptx.

* *Github Labels*: none

* *Status*: open

Issue 152 (https://github.com/netmod-wg/yang-next/issues/152)

YANG-next issue summary

Proposes a method to summarize and prioritize YANG-next issues by grouping and filtering by importance and complexity.

* *Github Labels*: none

* *Status*: open

Issue 153 (https://github.com/netmod-wg/yang-next/issues/153)

Allow ’when’ to be a child of the ’refine’ statement

Already if-feature and must statements are children under "refine", why not the "when" statement...?

* *Github Labels*: none

* *Status*: open

Issue 154 (https://github.com/netmod-wg/yang-next/issues/154)

Add ability to remove nodes from a grouping in a "uses" statement

This issue requests: Add ability to remove nodes from a grouping in a "uses" statement.

* *Github Labels*: none

* *Status*: open

Issue 155 (https://github.com/netmod-wg/yang-next/issues/155)

Allow hexadecimal notation for enum values

It would be beneficial to allow encoding of the enum value in hexadecimal, example: This is not allowed in YANG 1.1. Howere, default integer values are allowed to be encoded in hexadecimal.

* *Github Labels*: none

* *Status*: open

Issue 156 (https://github.com/netmod-wg/yang-next/issues/156)

canonical forms for strings (aka: the mac-address issue)

mac-address in both IETF and IEEE yang is defined as a string. The canonical representations are not the same.

* *Github Labels*: none

* *Status*: open

Author’s Address

Robert Wilton
Cisco
Email: rwilton@cisco.com