

YANGcore Pre-Alpha-1 Release Notes

Watsen Networks

February 15, 2025

Abstract

This documentation provides release notes for the Pre-Alpha-1 release of YANGcore.

Contents

1	Introduction	3
2	Implemented / Tested	3
3	Should Work	3
4	Upcoming Features/Releases:	4
4.1	Alpha	4
4.2	Beta	4
4.3	FCS	4
4.4	Post 1.0	5
5	Known Limitations	5
6	Change Log	6
6.1	0.0.1	6

1 Introduction

The sections below cover what is and is not working in this release of the YANGcore product.

Most everything listed as “not working” will be implemented before FCS.

2 Implemented / Tested

This section describes what is working in YANGcore. All of the following has been tested¹:

- RESTCONF API for the “native interface”².
- Other “facade” interfaces may be added by YANGcore-based applications³.
- Each listening NBI port is HTTP with or without TLS⁴.
- RESTCONF HEAD, GET, POST, PUT, and DELETE work over entire tree.
- Ordered-by user query parameters (‘insert’ and ‘point’) work.
- Pagination query parameters (‘limit’, ‘offset’, and ‘direction’) work.
- The ./well-known/host-meta, RESTCONF root (i.e., {+restconf}), and YANG-library resources.
- Tested using in-memory, file-base, MySQL and, a long time ago, AWS Aurora and Postgres.
- TLS connection to backend RDBMS, with or w/o client certificate.
- Plugin-based dynamic callouts to relay notifications
- Plugin-based dynamic callouts to relay audit log entries
- Logging and dynamic callouts for Audit log and Notification log
- Database-level transactions and concurrent access.
- Python 3.11 and 3.12

3 Should Work

This section regards things that should work, but haven’t been tested yet.

- IPv6: The default port binds to “127.0.0.1”, but this can be changed by setting the “YANG-CORE_DEFAULT_ADDR” environment variable. Other than this, there is no other IPv4-only code in the product as everything is handled by Python modules.
- UTF-8: Python string types support both ASCII and unicode, like UTF-8. Presumably string handling is ambivalent, but this isn’t tested yet.
- Windows: the software has been developed and tested exclusively in UNIX based systems. Python is very portable, and YANGcore has almost no interaction with the filesystem, so running on Windows might work, but this has not been tested.

¹There was, at one time, more than twice the number of lines of test code than code in YANGcore itself.

²Support for tenant interfaces will be added back later.

³e.g., the SZTPD application adds the “rfc8572-interface”.

⁴An external TLS-terminator must be used when a YANGcore listening port is configured to NOT use HTTPS.

4 Upcoming Features/Releases:

The following features are sorted by the expected release they might show up in. Please let us know if there is something out of place or missing.

4.1 Alpha

- plugin for external authenticator
- Support dynamic callout for authorization.
- Make access-control schema conditional.
- Implement proper RBAC: Noun - Verb - Target⁵.
- Email-based admin activations. It is intended that YANGcore will send email based notifications to newly created admins when their accounts are first created. It is important to validate the email address because the same email address may be emailed later when the password expiration date is approaching. This is why the email address is the primary key for the admin accounts.

4.2 Beta

- Run performance and soak tests. Only address issues found.

4.3 FCS

- Nothing new
- Stress tests
- Soak tests

⁵RBAC is currently an enum with values: “unrestricted”, “typical”, and “minimal”.

4.4 Post 1.0

In no particular order:

- Client certificate based auth to NBIs.
- Webhook-based callouts dynamic callouts. Some webhooks were implemented before, but better to re-build them generically on top of plugin-based callbacks. Workaround is for clients to use plugin-based callbacks.
- The <system> datastore, to share host-data with tenants⁶.
 - data includes some of these: plugins, device-types, preferences truststore, yang library
- init perf / scale / soak tests
- switch to sending notifications using draft-ietf-netconf-https-notif?
- Implicit change tracking. This is needed to support a few of the features listed below. It is also needed to detect when *any* part of the “transport” configuration changes, so that a SIGHUP can be issued. Currently SIGHUP is issued only when an “endpoint” is added or removed.
- Reference tracking. It is desired to track references to objects in the system. In YANG terms, these are the ‘leafref’ statements that reference a ‘list’ statement’s ‘key’ node. Tracking includes the current number of references and the time of last reference.
- Automated purging with notifications. It is desired to send notifications when unreferenced objects have been without a reference, for some configurable amount of time. A series of notifications with escalating urgency can be sent to configurable notification receivers. A final notification is sent when the purging occurs.
- Password minimum length constraints. Currently the “preference” setting for the admin account password’s minimum length constraint is ignored.
- Private key encryption. It is desirable for YANGcore’s Keystore implementation to have an ability to encrypt private keys via a “root” key. Note that this is above and beyond DB-level encryption; its purpose is to shield keys even from administrators having appropriate access. This root key would be protected by access control and/or an HSM.
- Actions: None of the ‘action’ statements defined in the YANG modules are implemented yet. Primarily this effects the ability to create keys, and generate certificate signing requests. The workaround is to generate the keys and certs using an external PKI and then pass those values in as configuration.

5 Known Limitations

- Python 3.13 isn’t supported, because of a failing pytest. Expectation is that the break in compatibility was accidental and will auto-resolve in a future 3.13 patch release.
- Due to issues with both Python and NGINX, when terminating TLS connections, limit client certificates to containing just the end-entity certificate (no intermediate certificates). It has been awhile since these issues were detected and hence may be resolved already.
- Python is (currently) unable to staple OSCP Responses to the TLS Handshake. Workaround is to front YANGcore with an external TLS Terminator (e.g., NGINX), which is better for performance anyway.
- RESTCONF layer doesn’t implement the “ietf-restconf-monitoring” module, per Section 9 of RFC 8040. This is needed to identify which capabilities (e.g., query params) and streams (e.g., the notifications) YANGcore supports. Workaround is for clients to read the YANGcore documentation to know what all YANGcore support.
- RESTCONF layer doesn’t return the “Last-Modified” HTTP header in HEAD and GET responses. The POST, PUT, DELETE, and PATCH handling logic doesn’t inspect request headers for the

⁶For when multi-tenancy turned on again.

“If-Modified-Since” and “If-Unmodified-Since” HTTP header fields. Support for timestamps is a SHOULD in RFC 8040, so this is not an RFC violation.

- RESTCONF layer doesn’t return the “ETag” header in HEAD and GET responses. The POST, PUT, DELETE, and PATCH handling logic doesn’t inspect request headers for the “If-Match”, and “If-None-Match” HTTP header fields. Support for Etag is a MUST in RFC 8040 only on the top-level datastore resource, and unspecified for inner-resources.
- RESTCONF layer doesn’t support the “depth” and “fields” query parameters. These query parameters are described by [Section 4.8 of RFC 8040](#) to reduce the amount of data returned to the client. RFC 8040 does not require these query parameters to be supported. Workaround is for the client to discard the unwanted parts of the response.
- RESTCONF layer doesn’t support the “content” query parameter. Required by RFC 8040. Low priority since currently YANGcore doesn’t support distinct “config true” values in the <operational> datastore (NMDA).
- RESTCONF layer doesn’t support the “PATCH” method. This a MUST in RFC 8040. Workaround is to use PUT.
- The SQLAlchemy layer doesn’t support the current major release of SQLAlchemy (2.x). The SQLAlchemy project released 2.0 recently. It was hoped that the migration to 2.0 would be quick, but that panned out to not be true (see [Migration Guide](#)). A dedicated effort is needed to update to the 2.x release. In the meanwhile, YANGcore’s ‘setup.cfg’ file contains the line ‘sqlalchemy<2.0.0’ to force use of the pre-2.0 code.

6 Change Log

6.1 0.0.1

- initial public release.
- Many nodes moved and/or renamed.
- Multi-tenancy removed (temporary).
- Significant update from the YANGcore embedded in SZTPD 0.0.15.
- YANG simplified: collapsed groupings from old multi-tenancy solution
- The data model is no longer a single namespace (related to collapsing the groupings).
- Added a Notifications log (/yangcore:notification-log)
- Documentation overhauled (more friendly and now only documents what works)