NEW gTLD PROGRAM: NEXT ROUND

gTLD

Deep dive on RST v2.0

Gavin Brown | ICANN82 | ccNSO TechDay

ICANN

Agenda

01 Background

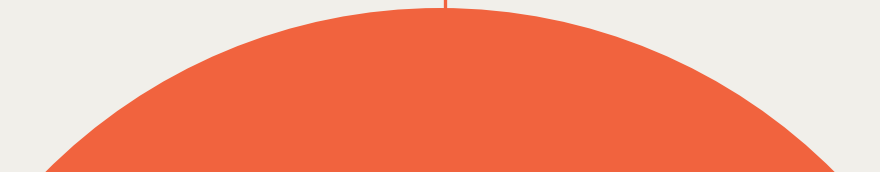
02 RST in the Next Round

03 API & Test Specifications

04 Development

05 Testing

06 Future plans



Background

- In the last round, the Pre-Delegation Testing (PDT) system was used to confirm the operational readiness of the critical functions of a TLD, prior to its delegation in the root zone.
- This system contained automated and manual evaluation components.
- Registries interacted by sending and receiving data through a case/ticket management system.
- Skilled engineers were required to run the tests. As a result, the rate at which TLDs could be tested was limited, and TLDs sometimes had to wait until a testing appointment was available.
- The PDT system was renamed to Registry System Testing (RST) and continues to be used to support existing gTLDs, when changing RSPs.

RST in the Next Round

- The Subsequent Procedures PDP recommended that “ICANN must develop a set of Registry System tests designed to demonstrate the technical capabilities of the registry operator” (Recommendation 39.1) and also that the RST system “must be efficient” (Recommendation 39.4).
- Given the limitations of the existing RST system, and the needs of the business both to support the Next Round and otherwise, the decision was made to develop a new system.
- The requirements for the new system were:
    1. The new system must only contain automated evaluation components.
    2. The new system must operate using a self-testing methodology minimizing the need for customer support from ICANN org.
    3. Registries must interact with RST through a RESTful API.
    4. In addition to Pre-Delegation Testing of Next Round gTLDs, the new system must also be able to support the Registry Service Provider (RSP) evaluation program, with specific test plans for each type of RSP applicant (main RSP, DNS RSP, DNSSEC RSP, and Proxy RSP).

Phase 1: Developing the test specifications

- Development of a new set of test specifications began in September 2023.
- The original PDT test specifications were used as the basis for the new test specifications.
- One innovation was the development of a machine-readable format for the test specifications, using YAML.
    - This meant that the *implementation* of RST v2.0 could be closely tied to the *specification*, so that (for example) test cases could be linked in a chain of dependencies, could not generate error messages that weren't specified for them, and so on.
    - The YAML version of the test specifications had a schema, which it could be validated against.
- For some test suites, (specifically DNS, DNSSEC and RDAP), external software components would be used for testing (Zonemaster for DNS and DNSSEC, and the RDAP Conformance Tool for RDAP). Tooling was developed to automatically import test case metadata from these tools into the test specifications.
- Tooling was written to convert the YAML version of the test specification into other formats (eg JSON) and into a human readable format, in HTML.
- A CI/CD pipeline allowed the formatting and publication of the test specs to GitHub Pages, with a weekly release schedule. Each release had a version number, and a ChangeLog described the changes between each release.

Phase 1: Developing the test specifications (ctd)

- The RST v2.0 test specifications has a hierarchical structure, consisting of:
  - *Test plans*, which are related to particular testing scenarios,
  - *Test suites*, which cover a particular element of the registry system, and
  - *Test cases*, which test a specific feature or behaviour of a system.
- Each test case is associated with one or more *input parameters*, and can generate one or more *error codes*.
- Test cases can also have *data providers*, which are tables of values used to test some aspect of the registry system, that can be changed without also changing the implementation.
- RST v2.0 tries to minimise the amount of up-front setup required by the Registry Operator or RSP. For EPP, RST v1.0 required the creation of various objects ahead of time, in order to test various commands (i.e. `<update>`, `<delete>`, `<transfer>`). In RST v2.0, the implementation will create these objects as needed, where possible. This reduces the preparation/reset time required to prepare for a test/retest.

Phase 1: Developing the test specifications (ctd)

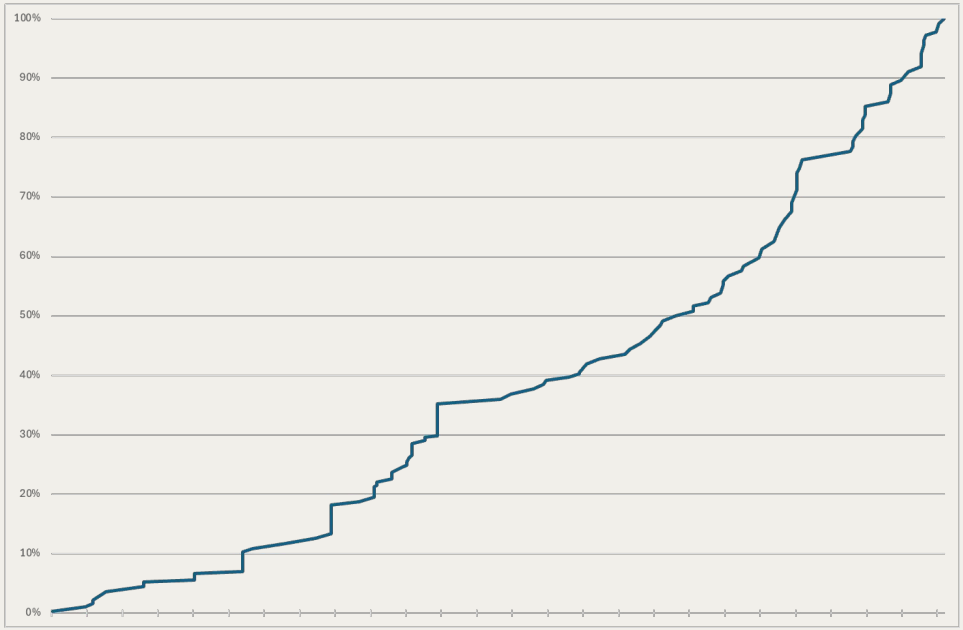
- The scope of the RST v2.0 test specifications is broader than that of RST v1.0, and also tests elements of the critical functions in greater depth.
- RST v2.0 introduces support for testing the following:
  - RDAP
  - DNSSEC algorithm rollovers
  - Minimum RPMs (support for the TMCH and the EPP Launch Extension)
- When converted to human-readable format, the RST v2.0 test specifications are over 650 pages in length, and more than 90,000 words.
- There are 11 test suites and 120 test cases, 516 error codes, and over 1,000 different error conditions (combination of test case and error code).
- The test specs (and API spec) were published for Public Comment in May 2024.

Phase 2: Developing the API specification

- In parallel to development of the test specifications, work also began on the specification of the RESTful API (RST-API).
- OpenAPI (sometimes referred to as Swagger) was used to describe this API.
- OpenAPI also uses YAML/JSON to provide machine-readable representations of the API.
- JSON Schema are used to describe JSON request/response payloads, URL and query parameters, etc.
- Tooling was developed to automatically import data elements from the test specifications (input parameters, error codes, etc).
- The use of OpenAPI leverages the tooling available to generate code for API clients and servers.
- The API spec had to be backported to v3.0 of the OpenAPI spec, due to the immaturity of support for v3.1 in code generators.
- As with the test specs, a CI/CD pipeline was used to publish machine- and human-readable versions of the API spec to a GitHub Pages site, with a weekly release schedule, and a Change Log.

Phase 3: implementation

- Implementing the RST v2.0 system was the longest and most challenging part of this project!
- Development began in January 2024, with a mixed team of ICANN staff and contractors, using an Agile methodology (Scrum).
- Test suites were converted to Epics, and test cases were converted to Stories within each Epic.
- As test cases and API endpoints were implemented, issues with the API and test specs were identified, and fixes incorporated into the next release.
- The development team's subject matter expertise grew substantially over the duration of the project, as can be seen in increase in velocity in the burnup chart:



Finding bugs…

- During the development of one EPP-related test case, we found a bug - in a twenty-year old RFC!
- The text of RFC 3915 describes how EPP responses contain a single <rgpStatus> element, but the XML schema allows for multiple elements.
- The test specification was based on the text of the RFC, but the implementation used the XML schema. When the developer raised the issue (as a potential bug in the test spec), we realised the RFC had an error.
- We also discovered that the version of the XML schema published by IANA (in the IETF XML Registry) was different to that in the RFC.
- This we reported to the RFC author and the regext mailing list, and as a result, an update to RFC 3915 may be published in the future.

Phase 4: testing the system

- The system is written in Java and uses Spring Boot. It contains over 72,000 lines of code.
- The test specs and API specs are imported into the code base and used to generate the basic components of the system.
- Integration testing was done using an open source domain registry system running in a cloud environment, combined with unit tests (jUnit) and Behaviour-Driven Development (Cucumber.io).
- The final test case was finished in early February 2025, and the focus of the project moved to testing.
- With over 1,000 error conditions, each of which requires a custom test harness (e.g. an EPP server that is broken in a specific way), this was always going to be a significant and complex undertaking, and is still under way.
- The team developed test tooling, including an EPP proxy server that can be configured to modify EPP commands and responses in-flight between client and server. This proxy was published as an open source Perl module (https://metacpan.org/pod/Net::EPP::MITMProxy).
- The Operational Testing & Evaluation (OT&E) environment was opened in November 2024, which allows current and prospective Registry Operators and RSPs to integrate RST into their own tooling, prepare for production RST during RSP evaluation, and validate the correct implementation of the test cases.

Operational Testing & Evaluation (OT&E)

- The OT&E system was launched in November 2024, with 70% test case coverage.
- As of February 2025, development of the test cases is complete, and the OT&E environment can be used with full set of test plans and test suites.
- In OT&E, you can create and configure test objects as needed, to support development and testing of registry systems. Users are limited to one concurrent test run (but any number of test objects).
- Access to the OT&E environment is available to current and prospective Registry Operators and RSPs.
- To get access:

  1. Generate a self-signed certificate, and a TLSA record for it;
  2. Publish the TLSA record in a DNSSEC-signed zone;
  3. Submit a request to ICANN (via globalsupport@icann.org or the NSp/RSP application system).

  Instructions for steps #1 and #2 may be found here:

  https://www.icann.org/resources/registry-system-testing-v2.0/#example

  If you have any questions, problems, or feedback, please contact globalsupport@icann.org.

Future plans

In the future, we hope to be able to:

- Release a command line client for interacting with the RST-API (production & OT&E)
- Release the test runner as open source software
- Implement DoT, DoH and DoQ for the “Additional DNS Transports” test suite

Further information

RST v2.0 page:

https://www.icann.org/resources/registry-system-testing-v2.0/

Test specs:

https://icann.github.io/rst-test-specs/

API spec:

https://icann.github.io/rst-test-specs/

Any questions?

Visit us at https://newgtldprogram.icann.org

- @icann
- facebook.com/icannorg
- youtube.com/icannnews
- flickr.com/icann
- linkedin.com/company/icann
- instagram.com/icannorg

ICANN