

Extensible Provisioning Protocol (EPP) mapping for DNS Time-To-Live (TTL) values



Gavin Brown, Principal Engineer, GDS Technical Services

ICANN78 TechDay/CPH TechOps
2023-10-22/23

Agenda

1. Problem Statement
2. TTL extension overview
3. Operational considerations for registries
4. Document status
5. Outstanding issue(s)
6. Q&A

Problem statement

- The TTL value on delegation records (NS, DS, glue) is decided by the registry, not the registrant, registrar, or DNS operator
- A wide array of values are in use, ranging from 1 hour up to 2 days (as in .com and .net)
- A long TTL means a long rollback period for changes
- A long rollback period deters rollout of “risky” changes such as DNSSEC
- Providing a way to allow TTL values above the zone cut to be changed prior to a change provides safety and will encourage deployment of (for example) DNSSEC

TTL extension overview

- The TTL extension adds new object properties to domain and host objects
- TTL values can be queried (<info>) and set/unset as part of <create> and <update> commands
- Two models:
 - a “global” TTL (applied to all records)
 - type-specific TTLs (e.g. for NS, DS, DNAME, A/AAAA records)
- Host objects inherit the TTL of any superordinate domain object

Examples

```
S:<?xml version="1.0" encoding="utf-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <domain:infData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:        <domain:name>example.com</domain:name>
S:        <domain:roid>EXAMPLE1-REP</domain:roid>
S:        <domain:status s="ok" />
S:      </domain:infData>
S:    </resData>
S:    <extension>
S:      <ttl:infData xmlns:ttl="urn:ietf:params:xml:ns:epp:ttl-1.0">
S:        <ttl:ttl>3600</ttl:ttl>
S:      </ttl:infData>
S:    </extension>
S:  </response>
S:</epp>
```

Examples

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <host:infData xmlns:host="urn:ietf:params:xml:ns:host-1.0">
S:        <host:name>ns1.example.com</host:name>
S:        <host:roid>NS1_EXAMPLE1-REP</host:roid>
S:        <host:status s="linked" />
S:        <host:addr ip="v6">1080::8:800:200C:417A</host:addr>
S:      </host:infData>
S:    </resData>
S:    <extension>
S:      <ttl:infData xmlns:ttl="urn:ietf:params:xml:ns:epp:ttl-1.0">
S:        <ttl:ttls>
S:          <ttl:A>86400</ttl:A>
S:          <ttl:AAAA>86400</ttl:AAAA>
S:        </ttl:ttls>
S:      </ttl:infData>
S:    </extension>
S:  </response>
S:</epp>
```

Examples

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <create>
C:      <host:create xmlns:host="urn:ietf:params:xml:ns:host-1.0">
C:        <host:name>ns1.example.com</host:name>
C:        <host:addr ip="v4">192.0.2.2</host:addr>
C:        <host:addr ip="v4">192.0.2.29</host:addr>
C:        <host:addr ip="v6">1080::8:800:200C:417A</host:addr>
C:      </host:create>
C:    </create>
C:    <extension>
C:      <ttl:create xmlns:ttl="urn:ietf:params:xml:ns:epp:ttl-1.0">
C:        <ttl:ttl>3600</ttl:ttl>
C:      </ttl:create>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

Operational considerations for registries

- As a general rule, lower TTLs results in higher query rates
- Higher TTLs mean changes take longer to be visible everywhere
- There is a “goldilocks” zone of “just right” TTL values
- Registry operators should restrict the TTL values they allow registrars to set
- Short TTLs may also make “fast flux” practices more effective

Document status

- Proprietary CentralNic TTL extension (~2006)
- Revived in draft-regext-brown-epp-ttl (July 2022)
- Adopted by the regext working group (May 2023)
- Latest version is draft-ietf-regext-epp-ttl-02 (September 2023)

Outstanding issues(s)

- It would be desirable to have an XML schema that can fully validate all client frames, however this leads to inelegant XML syntax where per-rrtype TTL values are enclosed in per-rrtype elements:

```
<ttl:NS>3600</ttl:NS>  
<ttl:DS>120</ttl:DS>
```

- A more elegant XML syntax is “non-deterministic” and requires business logic to validate client frames, adding risk and making implementation harder:

```
<ttl:ttn for="NS">3600</ttl:ttn>  
<ttl:ttn for="DS">120</ttl:ttn>
```

- Is there a way to have our cake and eat it? If not, do we want “pretty” or “safe”?
- Is the extension missing anything?

Questions, comments, feedback?



One World, One Internet

Visit us at **icann.org**



@icann



facebook.com/icannorg



youtube.com/icannnews



flickr.com/icann



linkedin/company/icann



soundcloud/icann



instagram.com/icannorg