Internet Engineering Task Force (IETF)
Request for Comments: 9803
Category: Standards Track
ISSN: 2070-1721

G. Brown
ICANN

June 2025

# Extensible Provisioning Protocol (EPP) Mapping for DNS Time-to-Live (TTL) Values

## Abstract

This document describes an extension to the Extensible Provisioning Protocol (EPP) that allows EPP clients to manage the Time-to-Live (TTL) value for domain name delegation records.

## Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9803.



## Table of Contents

## 1. Introduction

The principal output of any domain name registry system is a DNS zone file, which contains the delegation record(s) for names registered within a zone (such as a top-level domain). These records typically include one or more NS records, but may also include DS records for domains secured with DNSSEC [RFC9364], and DNAME records for Internationalized Domain Name (IDN) variants [RFC6927]. A and/or AAAA records may also be published for nameservers where they are required by DNS resolvers to avoid an infinite loop.

Typically, the Time-to-Live (TTL) value (see Section 5 of [RFC9499]) of these records is determined by the registry operator. However, in some circumstances it may be desirable to allow the sponsoring client of a domain name to change the TTL values used for that domain's delegation: for example, to reduce the amount of time required to complete a change of DNS servers, DNSSEC deployment or key rollover, or to allow for fast rollback of such changes.

This document describes an EPP extension to the domain name and host object mappings (described in [RFC5731] and [RFC5732], respectively) that allows the sponsor of a domain name or host object to change the TTL values of the resource record(s) associated with that object. It also describes how EPP servers should handle TTLs specified by EPP clients and how both parties coordinate to manage TTL values in response to changes in operational or security requirements.

### 1.1. Conventions Used in This Document

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.

In this document's examples, "C:" represents lines sent by a protocol client and "S:" represents lines returned by a protocol server. Indentation and white space in these examples are provided only to illustrate element relationships and are not required features of

this protocol.

A protocol client that is authorized to manage an existing object is described as a "sponsoring" client throughout this document.

XML is case sensitive. Unless stated otherwise, the XML specifications and examples provided in this document MUST be interpreted in the character case presented in order to develop a conforming implementation.

EPP uses XML namespaces to provide an extensible object management framework and to identify schemas required for XML instance parsing and validation. These namespaces and schema definitions are used to identify both the base protocol schema and the schemas for managed objects.

The XML namespace prefixes used in these examples (such as the string ttl in ttl:create) are solely for illustrative purposes. A conforming implementation MUST NOT require the use of these or any other specific namespace prefixes.

In accordance with Section 3.2.2.1 of XML Schema Part 2: Datatypes [XSD-DATATYPES], the allowable lexical representations for the xs:boolean datatype are the strings "0" and "false" for the concept 'false' and the strings "1" and "true" for the concept 'true'. Implementations MUST support both styles of lexical representation.

## 1.2. Extension Elements

This extension adds additional elements to the EPP domain and host mappings.

### 1.2.1. The <ttl:ttl> Element

The <ttl:ttl> element is used to define TTL values for the DNS resource records associated with domain and host objects.

<ttl:ttl> elements have the optional following attributes, depending on whether they appear in an EPP command or response:

"for"
: REQUIRED in both commands and responses, and specifies the DNS record type to which the TTL value pertains. This attribute MUST have one of the following values: "NS", "DS", "DNAME", "A", "AAAA" or "custom".

"custom"
: If the value of the "for" attribute is "custom", then the <ttl:ttl> element MUST also have a "custom" attribute containing a DNS record type conforming with the regular expression in Section 3.1 of [RFC6895]. Additionally, the record type MUST be registered with IANA in [IANA-RRTYPES].

"min"
: MUST NOT be present in EPP commands but MAY be present in EPP responses (see Section 2.1.1). It is used by the server to indicate the lowest value that may be set.

"default"
: MUST NOT be present in EPP commands but MAY be present in EPP responses (see Section 2.1.1). It is used by the server to indicate the default value.

"max"
: MUST NOT be present in EPP commands but MAY be present in EPP responses (see Section 2.1.1). It is used by the server to

indicate the highest value that may be set.

When present, the value of the "min" attribute MUST be lower than the value of the "max" attribute. The "default" attribute MUST be between the "min" and "max" values, inclusively.

#### 1.2.1.1. Element Content

The XML schema found in Section 8 of this document restricts the content of <ttl:ttl> elements to be either:

1. a non-negative integer, indicating the value of the TTL in seconds, or

2. empty, in which case the server's default TTL for the given record type is to be applied.

#### 1.2.1.2. Supported DNS Record Types

To facilitate forward compatibility with future changes to the DNS protocol, this document does not enumerate or restrict the DNS record types that can be included in the "custom" attribute of the <ttl:ttl> element.

The regular expression that is used to validate the values of the "custom" attribute is based on the expression found in Section 3.1 of [RFC6895], and it is intended to match both existing and future RRTYPE mnemonics. This eliminates the need to update this document in the event that new DNS records that exist above a zone cut (Section 7 of [RFC9499]) are specified.

Nevertheless, EPP servers that implement this extension MUST restrict the DNS record types that are accepted in <create> and <update> commands, and included in <info> responses, allowing only those types that are (a) registered in [IANA-RRTYPES] and (b) appropriate for use above a zone cut.

A server that receives a <create> or <update> command that attempts to set TTL values for inapplicable DNS record types MUST respond with a 2306 "Parameter value policy" error.

As an illustrative example, a server MAY allow clients to specify TTL values for the following record types for domain objects:

1. NS;

2. DS (if the server also implements [RFC5910]);

3. DNAME (if the server implements IDN variants using DNAME records).

##### 1.2.1.2.1. Glue Records

Glue records are described in Section 7 of [RFC9499].

Servers that implement host objects [RFC5732] MAY allow clients to specify TTL values for A and AAAA records for host objects.

A server supporting host objects that receives a command that attempts to set TTL values for A and AAAA records on a domain object MUST respond with a 2306 "Parameter value policy" error.

EPP servers that use the host attribute model (described in Section 1.1 of [RFC5731]) MAY allow clients to specify TTL values for A and AAAA records for domain objects.

#### 1.2.1.3. The <ttl:info> Element

The <ttl:info> element is used by clients to request that the server include additional information in <info> responses for domain and host objects.

It has a single OPTIONAL "policy" attribute, which takes a boolean value with a default value of "false".

The semantics of this element are described in Section 2.1.1.

Below is an example of a <ttl:info> element with an explicit "policy" attribute:

```
<ttl:info policy="true"/>
```

### 1.2.2. Examples

#### 1.2.2.1. Explicit TTL Value (<create> or <update> Command)

```
<ttl:ttl for="NS">3600</ttl:ttl>
```

#### 1.2.2.2. Explicit TTL Value (<info> Policy Mode)

```
<ttl:ttl
  for="NS"
  min="60"
  default="86400"
  max="172800">3600</ttl:ttl>
```

#### 1.2.2.3. Empty Value Indicating Default TTL (<create> or <update> Command, <info> Default Mode)

```
<ttl:ttl for="NS"/>
```

#### 1.2.2.4. Custom Record Type (<create> or <update> Command, <info> Default Mode)

```
<ttl:ttl
  for="custom"
  custom="NEWRRTYPE">3600</ttl:ttl>
```

## 2. EPP Command Mapping

### 2.1. EPP Query Commands

#### 2.1.1. EPP <info> Command

This extension defines an additional element for EPP <info> commands and responses for domain and host objects.

The EPP <info> command is extended to support two different modes:

1. The Default Mode (Section 2.1.1.1), which requests the inclusion of all non-default TTL values in the response; and

2. The Policy Mode (Section 2.1.1.2), which requests the inclusion of TTL information for all supported DNS record types in the response, along with the minimum, default, and maximum values for those records.

##### 2.1.1.1. Default Mode

If a server receives an <info> command for a domain or host object that includes a <ttl:info> element with a "policy" attribute that is "0" or "false", then the EPP response MUST contain <ttl:ttl> records

for all DNS record types that have non-default TTL values.  These elements MUST NOT have the "min", "default", and "max" attributes.

Below is an example domain <info> command with a <ttl:info> element with a "policy" attribute that is "false":

```
C: <?xml version="1.0" encoding="UTF-8" standalone="no"?>
C: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:   <command>
C:     <info>
C:       <domain:info
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:         <domain:name>example.com</domain:name>
C:       </domain:info>
C:     </info>
C:     <extension>
C:       <ttl:info
C:         xmlns:ttl="urn:ietf:params:xml:ns:epp:ttl-1.0"
C:         policy="false"/>
C:     </extension>
C:   </command>
C: </epp>
```

Below is an example domain <info> response to a command with a <ttl:info> element with a "policy" attribute that is "false":

```
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
S:         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:ns>
S:           <domain:hostObj>ns1.example.com</domain:hostObj>
S:           <domain:hostObj>ns1.example.net</domain:hostObj>
S:         </domain:ns>
S:         <domain:clID>ClientX</domain:clID>
S:         <domain:crID>ClientX</domain:crID>
S:         <domain:crDate>2023-11-08T10:14:55.0Z</domain:crDate>
S:         <domain:exDate>2024-11-08T10:14:55.0Z</domain:exDate>
S:       </domain:infData>
S:     </resData>
S:     <extension>
S:       <ttl:infData
S:         xmlns:ttl="urn:ietf:params:xml:ns:epp:ttl-1.0">
S:         <ttl:ttl for="NS">172800</ttl:ttl>
S:         <ttl:ttl for="DS">300</ttl:ttl>
S:       </ttl:infData>
S:       <secDNS:infData
S:         xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1">
S:         <secDNS:dsData>
S:           <secDNS:keyTag>12345</secDNS:keyTag>
S:           <secDNS:alg>13</secDNS:alg>
S:           <secDNS:digestType>2</secDNS:digestType>
S:           <secDNS:digest>49FD46E6C4B45C55D4AC</secDNS:digest>
S:         </secDNS:dsData>
S:       </secDNS:infData>
S:     </extension>
S:     <trID>
S:       <clTRID>ABC-12345</clTRID>
```

```
S:       <svTRID>54322-XYZ</svTRID>
S:     </trID>
S:   </response>
S: </epp>
```

Below is an example host <info> command with a <ttl:info> element with a "policy" attribute that is "false":

```
C: <?xml version="1.0" encoding="UTF-8" standalone="no"?>
C: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:   <command>
C:     <info>
C:       <host:info
C:        xmlns:host="urn:ietf:params:xml:ns:host-1.0">
C:         <host:name>ns1.example.com</host:name>
C:       </host:info>
C:     </info>
C:     <extension>
C:       <ttl:info
C:         xmlns:ttl="urn:ietf:params:xml:ns:epp:ttl-1.0"
C:         policy="false"/>
C:     </extension>
C:   </command>
C: </epp>
```

Below is an example host <info> response to a command with a <ttl:info> element with a "policy" attribute that is "false":

```
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
S:         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="ok"/>
S:         <host:addr ip="v4">192.0.2.2</host:addr>
S:         <host:addr ip="v6">2001:db8::8:800:200c:417a</host:addr>
S:         <host:clID>ClientX</host:clID>
S:         <host:crID>ClientX</host:crID>
S:         <host:crDate>2023-11-08T10:14:55.0Z</host:crDate>
S:       </host:infData>
S:     </resData>
S:     <extension>
S:       <ttl:infData
S:         xmlns:ttl="urn:ietf:params:xml:ns:epp:ttl-1.0">
S:         <ttl:ttl for="A">172800</ttl:ttl>
S:         <ttl:ttl for="AAAA">86400</ttl:ttl>
S:       </ttl:infData>
S:     </extension>
S:     <trID>
S:       <clTRID>ABC-12345</clTRID>
S:       <svTRID>54322-XYZ</svTRID>
S:     </trID>
S:   </response>
S: </epp>
```

#### 2.1.1.2. Policy Mode

If a server receives an <info> command for a domain or host object that includes a <ttl:info> element with a "policy" attribute that is "1" or "true", then the EPP response MUST contain <ttl:ttl> records

for all supported DNS record types, irrespective of whether those record types are actually in use by the object in question. These elements MUST have the "min", "default", and "max" attributes.

Below is an example domain <info> command requesting the server policies:

```
C: <?xml version="1.0" encoding="UTF-8" standalone="no"?>
C: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:   <command>
C:     <info>
C:       <domain:info
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:         <domain:name>example.com</domain:name>
C:       </domain:info>
C:     </info>
C:     <extension>
C:       <ttl:info
C:         xmlns:ttl="urn:ietf:params:xml:ns:epp:ttl-1.0"
C:         policy="true"/>
C:     </extension>
C:   </command>
C: </epp>
```

Below is an example domain <info> response providing the server policies:

```
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
S:         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:ns>
S:           <domain:hostObj>ns1.example.com</domain:hostObj>
S:           <domain:hostObj>ns1.example.net</domain:hostObj>
S:         </domain:ns>
S:         <domain:clID>ClientX</domain:clID>
S:         <domain:crID>ClientX</domain:crID>
S:         <domain:crDate>2023-11-08T10:14:55.0Z</domain:crDate>
S:         <domain:exDate>2024-11-08T10:14:55.0Z</domain:exDate>
S:       </domain:infData>
S:     </resData>
S:     <extension>
S:       <ttl:infData
S:         xmlns:ttl="urn:ietf:params:xml:ns:epp:ttl-1.0">
S:         <ttl:ttl for="NS"
S:           min="3600"
S:           default="86400"
S:           max="172800">172800</ttl:ttl>
S:         <ttl:ttl for="DS"
S:           min="60"
S:           default="86400"
S:           max="172800">300</ttl:ttl>
S:       </ttl:infData>
S:       <secDNS:infData
S:         xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1">
S:         <secDNS:dsData>
S:           <secDNS:keyTag>12345</secDNS:keyTag>
S:           <secDNS:alg>13</secDNS:alg>
```

```
S:           <secDNS:digestType>2</secDNS:digestType>
S:           <secDNS:digest>49FD46E6C4B45C55D4AC</secDNS:digest>
S:         </secDNS:dsData>
S:       </secDNS:infData>
S:     </extension>
S:     <trID>
S:       <clTRID>ABC-12345</clTRID>
S:       <svTRID>54322-XYZ</svTRID>
S:     </trID>
S:   </response>
S: </epp>
```

Below is an example host <info> command requesting the server policies:

```
C: <?xml version="1.0" encoding="UTF-8" standalone="no"?>
C: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:   <command>
C:     <info>
C:       <host:info
C:        xmlns:host="urn:ietf:params:xml:ns:host-1.0">
C:         <host:name>ns1.example.com</host:name>
C:       </host:info>
C:     </info>
C:     <extension>
C:       <ttl:info
C:         xmlns:ttl="urn:ietf:params:xml:ns:epp:ttl-1.0"
C:         policy="true"/>
C:     </extension>
C:   </command>
C: </epp>
```

Below is an example host <info> response providing the server policies:

```
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
S:         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="ok"/>
S:         <host:addr ip="v4">192.0.2.2</host:addr>
S:         <host:addr ip="v6">2001:db8::8:800:200c:417a</host:addr>
S:         <host:clID>ClientX</host:clID>
S:         <host:crID>ClientX</host:crID>
S:         <host:crDate>2023-11-08T10:14:55.0Z</host:crDate>
S:       </host:infData>
S:     </resData>
S:     <extension>
S:       <ttl:infData
S:         xmlns:ttl="urn:ietf:params:xml:ns:epp:ttl-1.0">
S:         <ttl:ttl for="A"
S:           min="3600"
S:           default="86400"
S:           max="172800">172800</ttl:ttl>
S:         <ttl:ttl for="AAAA"
S:           min="3600"
S:           default="86400"
S:           max="172800">86400</ttl:ttl>
S:       </ttl:infData>
```

```
S:     </extension>
S:     <trID>
S:       <clTRID>ABC-12345</clTRID>
S:       <svTRID>54322-XYZ</svTRID>
S:     </trID>
S:   </response>
S: </epp>
```

## 2.2. EPP Transform Commands

### 2.2.1. EPP <create> Command

This extension defines an additional element for EPP <create> commands for domain and host objects.

The <command> element of the <create> command MAY contain an <extension> element that MAY contain a <ttl:create> element. This element MUST contain one or more <ttl:ttl> records as described in Section 1.2.

If an EPP server receives a <create> command containing a TTL value that is outside the server's permitted range, it MUST reject the command with a 2004 "Parameter value range error" response.

Below is an example domain <create> command:

```
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:       <domain:create
C:         xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:         <domain:name>example.com</domain:name>
C:         <domain:period unit="y">1</domain:period>
C:         <domain:ns>
C:           <domain:hostObj>ns1.example.com</domain:hostObj>
C:           <domain:hostObj>ns1.example.net</domain:hostObj>
C:         </domain:ns>
C:         <domain:authInfo>
C:           <domain:pw/>
C:         </domain:authInfo>
C:       </domain:create>
C:     </create>
C:     <extension>
C:       <ttl:create
C:         xmlns:ttl="urn:ietf:params:xml:ns:epp:ttl-1.0">
C:         <ttl:ttl for="NS">172800</ttl:ttl>
C:         <ttl:ttl for="DS">300</ttl:ttl>
C:       </ttl:create>
C:       <secDNS:create
C:         xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1">
C:         <secDNS:dsData>
C:           <secDNS:keyTag>12345</secDNS:keyTag>
C:           <secDNS:alg>13</secDNS:alg>
C:           <secDNS:digestType>2</secDNS:digestType>
C:           <secDNS:digest>49FD46E6C4B45C55D4AC</secDNS:digest>
C:         </secDNS:dsData>
C:       </secDNS:create>
C:     </extension>
C:     <clTRID>ABC-12345</clTRID>
C:   </command>
C: </epp>
```

Below is an example host <create> command:

```
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
C:         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="v6">2001:db8::8:800:200c:417a</host:addr>
C:       </host:create>
C:     </create>
C:     <extension>
C:       <ttl:create
C:         xmlns:ttl="urn:ietf:params:xml:ns:epp:ttl-1.0">
C:         <ttl:ttl for="A"/>
C:         <ttl:ttl for="AAAA">86400</ttl:ttl>
C:       </ttl:create>
C:     </extension>
C:     <clTRID>ABC-12345</clTRID>
C:   </command>
C: </epp>
```

### 2.2.2. EPP <update> Command

This extension defines an additional element for EPP <update> commands for domain and host objects.

The <command> element of the <update> command MAY contain an <extension> element that MAY contain a <ttl:update> element. This element MUST contain one or more <ttl:ttl> records as described in Section 1.2.

If an EPP server receives an <update> command containing a TTL value that is outside the server’s permitted range, it MUST reject the command with a 2004 "Parameter value range error" response.

Below is an example domain <update> command:

```
C: <?xml version="1.0" encoding="UTF-8" standalone="no"?>
C: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:   <command>
C:     <update>
C:       <domain:update
C:         xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:         <domain:name>example.com</domain:name>
C:       </domain:update>
C:     </update>
C:     <extension>
C:       <ttl:update
C:         xmlns:ttl="urn:ietf:params:xml:ns:epp:ttl-1.0">
C:         <ttl:ttl for="NS"/>
C:         <ttl:ttl for="custom"
C:           custom="DELEG"/>
C:         <ttl:ttl for="DS">86400</ttl:ttl>
C:       </ttl:update>
C:     </extension>
C:     <clTRID>ABC-12345</clTRID>
C:   </command>
C: </epp>
```

Below is an example host <update> command:

```
C: <?xml version="1.0" encoding="UTF-8" standalone="no"?>
C: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:   <command>
C:     <update>
C:       <host:update
```

```
C:         xmlns:host="urn:ietf:params:xml:ns:host-1.0">
C:         <host:name>ns1.example.com</host:name>
C:       </host:update>
C:     </update>
C:     <extension>
C:       <ttl:update
C:         xmlns:ttl="urn:ietf:params:xml:ns:epp:ttl-1.0">
C:         <ttl:ttl for="A">86400</ttl:ttl>
C:         <ttl:ttl for="AAAA">3600</ttl:ttl>
C:       </ttl:update>
C:     </extension>
C:     <clTRID>ABC-12345</clTRID>
C:   </command>
C: </epp>
```

## 3. Server Processing of TTL Values

### 3.1. Permitted Record Types

EPP servers MAY restrict the supported DNS record types. For example, a server MAY allow clients to specify TTL values for DS records only.

A server that receives a <create> or <update> command that includes a restricted record type MUST respond with a 2306 "Parameter value policy" error.

Clients can discover the DNS record types for which an EPP server permits TTL values to be changed by performing a Policy Mode <info> command, as outlined in Section 2.1.1.2.

### 3.2. Use of TTL Values in Delegation Records

EPP servers that implement this extension SHOULD use the values provided by EPP clients for the TTL values of records published in the DNS for domain and (if supported) host objects. Server operators MAY disregard these values in order to address security and stability issues, as described in Section 5 and Section 6.

EPP servers that use the host attribute model SHOULD use any NS, A, and/or AAAA TTL values specified for the domain object when publishing NS, A, and/or AAAA records derived from host attributes.

## 4. Out-of-Band Changes to TTL Values

In order to address operational or security issues, EPP server operators MAY make changes to TTL values out-of-band (that is, not in response to an <update> command received from the sponsoring client).

Server operators MAY also implement automatic reset of TTL values, so that they revert to the default value a certain amount of time after an update has been made.

If a TTL value is changed out-of-band, EPP server operators MAY notify the sponsoring client using the EPP Change Poll Extension [RFC8590], which provides a generalized method for EPP servers to notify clients of changes to objects under their sponsorship.

## 5. Operational Considerations

### 5.1. Operational Impact of TTL Values

Registry operators must consider the balance between registrants' desire for changes to domains to be visible in the DNS quickly, and the increased DNS query traffic that short TTLs can bring.

Registry operators SHOULD implement limits on the maximum and minimum accepted TTL values that are narrower than the values permitted in the XML schema in Section 8 (which were chosen to allow any TTL permitted in DNS records). This is in order to prevent scenarios where an excessively high or low TTL causes operational issues on either side of the zone cut.

Section 4 describes how server operators MAY unilaterally change TTL values in order to address operational or security issues, or only permit changes for limited time periods (after which TTLs revert to the default).

### 5.2. When TTL Values Should Be Changed

A common operational mistake is changing the DNS record TTLs during or after the planned change to the records themselves. This arises due to a misunderstanding about how TTLs work.

It is RECOMMENDED that guidance be provided to users so they are aware that changes to a TTL are only effective in shortening transition periods if implemented a period of time (at least equal to the current TTL) _before_ the planned change. The latency between receipt of the <update> command and the actual publication of the changes in the DNS should also be taken into consideration in this calculation.

### 5.3. Changes to Server Policy

Registry operators may change their policies relating to TTL values from time to time. Previously configured TTL values may consequently fall outside a newly applied policy. This document places no obligation on EPP server operators in respect of these values, and server operators may, as part of a policy change, change the TTL values specified by clients for domain and host objects. Section 4 describes how such out-of-band changes should be carried out.

## 6. Security Considerations

### 6.1. Fast Flux DNS

Some malicious actors use a technique called "fast flux DNS" [SAC-025] to rapidly change the DNS configuration for a zone in order to evade takedown and law enforcement activity. Server operators should take this into consideration when setting the lower limit on TTL values, since a short TTL on delegations may enhance the effectiveness of fast flux techniques on evasion.

Client implementations that provide an interface for customers to configure TTL values for domain names should consider implementing controls to deter and mitigate abusive behavior, such as those outlined in the "Current and Possible Mitigation Alternatives" section of [SAC-025].

### 6.2. Compromised User Accounts

An attacker who obtains access to a customer account at a domain registrar that supports this extension could make unauthorized changes to the NS and/or glue records for a domain, and then increase the associated TTLs so that the changes persist in caches for a long time after the attack has been detected.

Client implementations that provide an interface for customers to configure TTL values for domain names should consider implementing upper limits in order to reduce the impact of account compromise, in addition to best practices relating to credential management, multi-factor authentication, risk-based access control, and so on.

## 7. IANA Considerations

### 7.1. XML Namespace

This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism described in [RFC3688]. The following URI assignments have been made by IANA:

Registration for the TTL namespace:

URI: urn:ietf:params:xml:ns:epp:ttl-1.0
Registrant Contact: IESG
XML: None. Namespace URIs do not represent an XML specification.

Registration for the TTL XML schema:

URI: urn:ietf:params:xml:schema:epp:ttl-1.0
Registrant Contact: IESG
XML: See Section 8 of this document.

### 7.2. EPP Extension Registry

The EPP extension described in this document has been registered by IANA in the "Extensions for the Extensible Provisioning Protocol (EPP)" registry described in [RFC7451]. The details of the registration are as follows:

Name of Extension: Extensible Provisioning Protocol (EPP) Mapping for DNS Time-to-Live (TTL) Values
Document Status: Standards Track
Reference: RFC 9803
Registrant: IESG
TLDs: Any
IPR Disclosure: None
Status: Active
Notes: None

## 8. Formal Syntax

The formal syntax presented here is a complete schema representation of the extension suitable for automated validation of EPP XML instances.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
  xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="urn:ietf:params:xml:ns:epp:ttl-1.0"
  xmlns:ttl="urn:ietf:params:xml:ns:epp:ttl-1.0"
  elementFormDefault="qualified">
  <annotation>
    <documentation>
      Extensible Provisioning Protocol v1.0 extension
      schema for Time-to-Live (TTL) Values for domain
      and host objects.
    </documentation>
  </annotation>

  <element name="info">
    <complexType>
      <attribute name="policy" type="boolean" default="false"/>
    </complexType>
  </element>

  <!--
    <ttl> elements can appear in <create> and
```

```
  <update> commands, and <info> responses
-->

<element name="create" type="ttl:commandContainer">
  <unique name="uniqueRRTypeForCreate">
    <selector xpath="ttl:ttl"/>
    <field xpath="@for"/>
  </unique>
</element>

<element name="update" type="ttl:commandContainer">
  <unique name="uniqueRRTypeForUpdate">
    <selector xpath="ttl:ttl"/>
    <field xpath="@for"/>
  </unique>
</element>

<element name="infData" type="ttl:responseContainer">
  <unique name="uniqueRRTypeForInfo">
    <selector xpath="ttl:ttl"/>
    <field xpath="@for"/>
  </unique>
</element>

<complexType name="commandContainer">
  <sequence>
    <element
      name="ttl"
      type="ttl:commandTTLType"
      minOccurs="1"
      maxOccurs="unbounded"/>
  </sequence>
</complexType>

<complexType name="responseContainer">
  <sequence>
    <element
      name="ttl"
      type="ttl:responseTTLType"
      minOccurs="1"
      maxOccurs="unbounded"/>
  </sequence>
</complexType>

<complexType name="commandTTLType">
  <simpleContent>
    <extension base="ttl:ttlOrNull">
      <attribute
        name="for"
        type="ttl:rrType"
        use="required"/>

      <attribute
        name="custom"
        type="ttl:customRRType"/>
    </extension>
  </simpleContent>
</complexType>

<complexType name="responseTTLType">
  <simpleContent>
    <extension base="ttl:ttlOrNull">
      <attribute
        name="for"
        type="ttl:rrType"
        use="required"/>
```

```
        <attribute
          name="custom"
          type="ttl:customRRType"/>

        <attribute
          name="min"
          type="ttl:ttlValue"/>

        <attribute
          name="default"
          type="ttl:ttlValue"/>

        <attribute
          name="max"
          type="ttl:ttlValue"/>
      </extension>
    </simpleContent>
  </complexType>

  <!--
    union type allowing the element to either contain
    nothing or a TTL value
  -->
  <simpleType name="ttlOrNull">
    <union
      memberTypes="ttl:emptyValue ttl:ttlValue"/>
  </simpleType>

  <!-- empty value type -->
  <simpleType name="emptyValue">
    <restriction base="token">
      <length value="0"/>
    </restriction>
  </simpleType>

  <!-- TTL value type -->
  <simpleType name="ttlValue">
    <restriction base="nonNegativeInteger">
      <minInclusive value="0"/>
      <maxInclusive value="2147483647"/>
    </restriction>
  </simpleType>

  <!-- resource record mnemonic type -->
  <simpleType name="rrType">
    <restriction base="token">
      <enumeration value="NS" />
      <enumeration value="DS" />
      <enumeration value="DNAME" />
      <enumeration value="A" />
      <enumeration value="AAAA" />
      <enumeration value="custom" />
    </restriction>
  </simpleType>

  <!-- custom resource record type -->
  <simpleType name="customRRType">
    <restriction base="token">
      <pattern value="A|[A-Z][A-Z0-9\-]*[A-Z0-9]"/>
    </restriction>
  </simpleType>
</schema>
```

## 9. References

## 9.1. Normative References


[IANA-RRTYPES]
IANA, "Resource Record (RR) TYPEs", <https://www.iana.org/assignments/dns-parameters>.

[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/info/rfc2119>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[RFC5731] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Domain Name Mapping", STD 69, RFC 5731, DOI 10.17487/RFC5731, August 2009, <https://www.rfc-editor.org/info/rfc5731>.

[RFC5732] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Host Mapping", STD 69, RFC 5732, DOI 10.17487/RFC5732, August 2009, <https://www.rfc-editor.org/info/rfc5732>.

[RFC5910] Gould, J. and S. Hollenbeck, "Domain Name System (DNS) Security Extensions Mapping for the Extensible Provisioning Protocol (EPP)", RFC 5910, DOI 10.17487/RFC5910, May 2010, <https://www.rfc-editor.org/info/rfc5910>.

[RFC6895] Eastlake 3rd, D., "Domain Name System (DNS) IANA Considerations", BCP 42, RFC 6895, DOI 10.17487/RFC6895, April 2013, <https://www.rfc-editor.org/info/rfc6895>.

[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/info/rfc8174>.

[XSD-DATATYPES]
Biron, P., Ed. and A. Malhotra, Ed., "XML Schema Part 2: Datatypes Second Edition", W3C Recommendation, October 2004, <https://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>. Latest version available at <https://www.w3.org/TR/xmlschema-2/>.


## 9.2. Informative References


[RFC6927] Levine, J. and P. Hoffman, "Variants in Second-Level Names Registered in Top-Level Domains", RFC 6927, DOI 10.17487/RFC6927, May 2013, <https://www.rfc-editor.org/info/rfc6927>.

[RFC7451] Hollenbeck, S., "Extension Registry for the Extensible Provisioning Protocol", RFC 7451, DOI 10.17487/RFC7451, February 2015, <https://www.rfc-editor.org/info/rfc7451>.

[RFC8590] Gould, J. and K. Feher, "Change Poll Extension for the Extensible Provisioning Protocol (EPP)", RFC 8590, DOI 10.17487/RFC8590, May 2019, <https://www.rfc-editor.org/info/rfc8590>.

[RFC9364] Hoffman, P., "DNS Security Extensions (DNSSEC)", BCP 237, RFC 9364, DOI 10.17487/RFC9364, February 2023, <https://www.rfc-editor.org/info/rfc9364>.

[RFC9499] Hoffman, P. and K. Fujiwara, "DNS Terminology", BCP 219, RFC 9499, DOI 10.17487/RFC9499, March 2024, <https://www.rfc-editor.org/info/rfc9499>.

[SAC-025] ICANN Security and Stability Advisory Committee (SSAC), "SSAC Advisory on Fast Flux Hosting and DNS", SAC 025, January 2008, <https://www.icann.org/en/system/files/files/sac-025-en.pdf>.

## Acknowledgments

The author wishes to thank the following people for their advice and feedback during the development of this document:

* James Gould
* Hugo Salgado
* Patrick Mevzek
* Rick Wilhelm
* Marc Groeneweg
* Ties de Kock
* Tim Wicinski
* Jasdip Singh

## Author’s Address

Gavin Brown
ICANN
12025 Waterfront Drive, Suite 300
Los Angeles, CA 90292
United States of America
Email: gavin.brown@icann.org
URI: https://www.icann.org/