

Internet Engineering Task Force (IETF)
Internet-Draft
Intended status: Standards Track
Expires: June 30, 2021

T. Sattler
R. Carney
J. Kolker
GoDaddy Inc.
December 8, 2020

**Registry Maintenance Notifications for the
Extensible Provisioning Protocol (EPP)
draft-ietf-regext-epp-registry-maintenance-06**

Abstract

This document describes an Extensible Provision Protocol (EPP) mapping for registry's maintenance notifications.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress".

This Internet-Draft will expire on June 30, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
1.1.	Terminology and Definitions	3
2.	Object Attributes	3
2.1.	Internationalized Domain Names	3
2.2.	Dates and Times	3
2.3.	Maintenance Elements	4
3.	EPP Command Mapping	6
3.1.	EPP Query Commands	6
3.1.1.	EPP <check> Command	6
3.1.2.	EPP <transfer> Command	6
3.1.3.	EPP <info> Command	6
3.1.3.1.	Info Maintenance Item	6
3.1.3.2.	Info Maintenance List	8
3.1.4.	EPP <poll> Command	9
3.2.	EPP Transform Commands	11
3.2.1.	EPP <create> Command	11
3.2.2.	EPP <delete> Command	11
3.2.3.	EPP <renew> Command	11
3.2.4.	EPP <transfer> Command	11
3.2.5.	EPP <update> Command	11
4.	Formal Syntax	12
4.1.	Registry Maintenance EPP Mapping Schema	12
5.	IANA Considerations	17
5.1.	XML Namespace	17
5.2.	EPP Extension Registry	17
6.	Security Considerations	18
7.	Implementation Status	18
7.1.	GoDaddy Registry	18
8.	References	19
8.1.	Normative References	19
8.2.	Informative References	19
Appendix A.	Change History	19
A.1.	Change from draft-sattler-epp-poll-maintenance-response to draft-sattler-epp-registry-maintenance	19
A.2.	Change from draft-sattler-epp-registry-maintenance to draft-ietf-regext-epp-registry-maintenance	19
A.3.	Change from 00 to 01	19
A.4.	Change from 01 to 02	20
A.5.	Change from 02 to 03	20
A.6.	Change from 03 to 04	20
A.7.	Change from 04 to 05	20
A.8.	Change from 05 to 06	20
	Acknowledgments	21
	Authors' Addresses	21

[1. Introduction](#)

Registries usually conduct maintenances and inform registrars in different ways. Given the expansion of the DNS namespace, it is now desirable to provide a method for EPP servers to notify EPP clients as well as a method for EPP clients to query EPP servers for upcoming maintenances.

This document describes an extension mapping for version 1.0 of the Extensible Provision Protocol [[RFC5730](#)]. This mapping provides a mechanism by which EPP servers may notify and EPP clients to query for upcoming maintenances.

[1.1. Terminology and Definitions](#)

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[RFC2119](#)] when specified in their uppercase forms.

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

"maint" is used as an abbreviation for "urn:ietf:params:xml:ns:epp:maintenance-0.1". The XML namespace prefix "maint" is used, but implementations MUST NOT depend on it and instead employ a proper namespace-aware XML parser and serializer to interpret and output the XML documents.

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

[2. Object Attributes](#)

[2.1. Internationalized Domain Names](#)

Names of affected hosts MUST be provided as A-label according to [[RFC5891](#)].

[2.2. Dates and Times](#)

All dates and times attribute values MUST be expressed in Universal Coordinated Time (UTC) using the Gregorian calendar. The extended date-time form using upper case "T" and "Z" characters defined in ISO 8601 [[RFC3339](#)] MUST be used to represent date-time values.

2.3. Maintenance Elements

The `<maint:item>` element describes a single registry maintenance event during a specific period. This element is used in a maintenance item EPP `<info>` response and `<poll>` message.

For creating a new maintenance the attribute `<maint:crDate>` MUST be set and the attribute `<maint:upDate>` SHALL NOT be present.

For updating a maintenance the attributes `<maint:crDate>` and `<maint:upDate>` MUST be set.

`<maint:id>`

MUST be present and a server unique id and SHALL NOT be changed if maintenance is updated or deleted. A human-readable description of the maintenance is identified via an OPTIONAL "msg" attribute.

`<maint:systems>`

MUST be present and contains one or more `<maint:system>` elements. The server SHOULD NOT list systems which are not affected by the maintenance.

`<maint:system>`

MUST be present at least once and has an element of `<maint:name>`, `<maint:host>` and `<maint:impact>`.

`<maint:name>`

MUST be present and indicates the name of the affected system, such as "EPP", "WHOIS", "DNS", "Portal", etc.

`<maint:host>`

MUST be present and indicates the affected maintained system contains the hostname and OPTIONAL an IP address.

Hostname SHALL be A-label according [[RFC5891](#)].

IPv4 addresses SHALL be dotted-decimal notation.

An example of this textual representation is "192.0.2.0".

IPv6 addresses SHALL be according [[RFC5952](#)].

An example of this textual representation is "2001:db8::1:0:0:1".

`<maint:impact>`

MUST be present and contains the impact level; values MUST either be "full" or "partial".

`<maint:environment>`

MUST be present and indicates the type of the affected system; the attribute type is REQUIRED and SHOULD either be "production", "ote", "staging", "dev" or "custom". And alternatively the attribute name could be used to define a server specific affected

system for example. In that case name MUST be set:
<maint:environment type="custom" name="marketing"/>

<maint:start>

SHOULD be present and indicates the start of the maintenance according ISO 8601 [[RFC3339](#)].
Format: YYYY-MM-DDThh:mm:ssTZ

<maint:end>

SHOULD be present and indicates the end of the maintenance according to ISO 8601 [[RFC3339](#)], and MUST be equal to or greater than <maint:start>.
Format: YYYY-MM-DDThh:mm:ssTZ

<maint:reason>

MUST be present and contains the reason behind the maintenance; values SHOULD either be "planned" or "emergency".

<maint:detail>

MAY be present and contains URI to detailed maintenance description.

<maint:description>

MAY be present and provides a freeform description of the maintenance without having to create and traverse an external resource.

<maint:tlds>

SHOULD be present and contains <maint:tld> elements.

<maint:tld>

MUST be present and contains the affected top-level domain.
A-label format according to [[RFC5891](#)].

<maint:intervention>

SHOULD be present and contains <maint:connection> and <maint:implementation>.

<maint:connection>

MUST be present and indicates if a client needs to do something that is connection-related, such as a reconnect. The value SHALL be boolean.

<maint:implementation>

MUST be present and indicates if a client needs to do something that is implementation-related, such as a code change. The value SHALL be boolean.

<maint:crDate>

MUST be present and contains the creation date of the maintenance according ISO 8601 [[RFC3339](#)].
Format: YYYY-MM-DDThh:mm:ssTZ

<maint:update>

MAY be present and contains the updated date of the maintenance according to ISO 8601 [[RFC3339](#)], and if set MUST be equal to or greater than <main:crDate>.

Format: YYYY-MM-DDThh:mm:ssTZ

3. EPP Command Mapping

A detailed description of the EPP syntax and semantics can be found in the EPP core protocol specification [[RFC5730](#)]. The command mappings described here are specifically for the use to notify of Registry Maintenances and Registry Maintenance object mapping.

3.1. EPP Query Commands

EPP [[RFC5730](#)] provides three commands to retrieve object information: <check> to determine if an object is known to the server, <info> to retrieve detailed information associated with an object, and <transfer> to retrieve object transfer status information.

3.1.1. EPP <check> Command

Available check semantics do not apply to maintenance objects, so there is no mapping defined for the EPP <check> command.

3.1.2. EPP <transfer> Command

Transfer semantics do not apply to maintenance objects, so there is no mapping defined for the EPP <transfer> command.

3.1.3. EPP <info> Command

EPP provides the <info> command that is used to retrieve registry maintenance information. In addition to the standard EPP command elements, the <info> command MUST contain a <maint:info> element that identifies the maintenance namespace.

The <maint:info> element MUST contain a child element. It is either <maint:id> to retrieve a specific maintenance item or <maint:list> to query all maintenance items.

3.1.3.1. Info Maintenance Item

The information on a specific Maintenance Item can be retrieved by using <info> command with the element <maint:info> and the child element <maint:id>.

If a <maint:info> is sent with a <maint:id> that does not exist on the server side, then the result code 2303 MUST be responded.

Example to retrieve a specific maintenance item in an <info> command.

Sattler, et al.

Expires June 30, 2021

[Page 6]

```
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:      <maint:info
C:        xmlns:maint="urn:ietf:params:xml:ns:epp:maintenance-0.1">
C:          <maint:id>2e6df9b0-4092-4491-bcc8-9fb2166dcee6</maint:id>
C:        </maint:info>
C:      </info>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

When an `<info>` command has been processed successfully, the EPP `<resData>` element MUST contain a child `<maint:infData>` element that identifies the maintenance namespace. The `<maint:infData>` element contains the `<maint:item>` element defined in [Section 2.3](#).

Example of returning a specific maintenance item in an `<info>` response.

```
S:<?xml version="1.0" encoding="UTF-8"?>
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:      <maint:infData
S:        xmlns:maint="urn:ietf:params:xml:ns:epp:maintenance-0.1">
S:        <maint:item>
S:          <maint:id>2e6df9b0-4092-4491-bcc8-9fb2166dcee6
S:          </maint:id>
S:          <maint:systems>
S:            <maint:system>
S:              <maint:name>EPP</maint:name>
S:              <maint:host>epp.registry.example
S:            </maint:host>
S:              <maint:impact>full</maint:impact>
S:            </maint:system>
S:          </maint:systems>
S:          <maint:environment type="production"/>
S:          <maint:start>2020-09-30T06:00:00Z</maint:start>
S:          <maint:end>2020-09-30T14:25:57Z</maint:end>
```

```
S:      <maint:reason>planned</maint:reason>
S:      <maint:detail>
S:        https://www.registry.example/notice?123
S:      </maint:detail>
S:      <maint:description lang="en">free text
S:    </maint:description>
S:    <maint:tlds>
S:      <maint:tld>example</maint:tld>
S:      <maint:tld>test</maint:tld>
S:    </maint:tlds>
S:    <maint:intervention>
S:      <maint:connection>false</maint:connection>
S:      <maint:implementation>false</maint:implementation>
S:    </maint:intervention>
S:    <maint:crDate>2020-03-08T22:10:00Z</maint:crDate>
S:  </maint:item>
S: </maint:infData>
S: </resData>
S: <trID>
S:   <clTRID>ABC-12345</clTRID>
S:   <svTRID>54321-XYZ</svTRID>
S: </trID>
S: </response>
S: </epp>
```

3.1.3.2. Info Maintenance List

The Info Maintenance List can be retrieved by using the `<info>` command and the element `<maint:info>` with the child element `<maint:list/>`. Server policy determines if previous maintenances will be listed in the `<maint:list>`.

Example to retrieve the maintenance list in an `<info>` 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:     <info>
C:       <maint:info
C:         xmlns:maint="urn:ietf:params:xml:ns:epp:maintenance-0.1">
C:           <maint:list/>
C:         </maint:info>
C:       </info>
C:     <clTRID>ABC-12345</clTRID>
C:   </command>
C: </epp>
```

When an <info> command has been processed successfully, the EPP <resData> element MUST contain a child <maint:infData> element that identifies the maintenance namespace. The <maint:infData> element contains the <maint:list> element a list of <maint:listItem> elements. The <maint:listItem> element contains the following child elements:

- <maint:id>: <maint:id> defined in [Section 2.3](#).
- <maint:start>: <maint:start> defined in [Section 2.3](#).
- <maint:end>: <maint:end> defined in [Section 2.3](#).
- <maint:crDate>: <maint:crDate> defined in [Section 2.3](#).
- <maint:update>: OPTIONAL <maint:update> defined in [Section 2.3](#).

Example of returning the maintenance list in an <info> response.

```
S:<?xml version="1.0" encoding="UTF-8"?>
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:      <maint:infData
S:        xmlns:maint="urn:ietf:params:xml:ns:epp:maintenance-0.1">
S:        <maint:list>
S:          <maint:listItem>
S:            <maint:id>2e6df9b0-4092-4491-bcc8-9fb2166dcee6
S:          </maint:id>
S:            <maint:start>2020-04-30T06:00:00Z</maint:start>
S:            <maint:end>2020-04-30T07:00:00Z</maint:end>
S:            <maint:crDate>2020-02-08T22:10:00Z</maint:crDate>
S:          </maint:listItem>
S:          <maint:listItem>
S:            <maint:id>91e9dabf-c4e9-4c19-a56c-78e3e89c2e2f
S:          </maint:id>
S:            <maint:start>2020-06-15T04:30:00Z</maint:start>
S:            <maint:end>2020-06-15T05:30:00Z</maint:end>
S:            <maint:crDate>2020-02-08T22:10:00Z</maint:crDate>
S:            <maint:update>2020-03-08T20:11:00Z</maint:update>
S:          </maint:listItem>
S:        </maint:list>
S:      </maint:infData>
S:    </resData>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

3.1.4. EPP <poll> Command

The EPP <poll> command and response is defined in [Section 2.9.2.3 of \[RFC5730\]](#). The Registry Maintenance Notification is included in the EPP <poll> response of [\[RFC5730\]](#).

For the Registry Maintenance Notification, there are three types of poll messages. The poll message applies whenever the domain name registry creates, updates, or deletes maintenance. In the case of a Registry Maintenance specific message, a `<maint:infData>` element will be included within the `<resData>` element of the standard `<poll>` response.

The `<maint:infData>` element will include a reference to the Registry Maintenance namespace. EPP data contained within the `<maint:infData>` element is formatted according to the maintenance-poll schema.

Please see the definition of `<maint>` elements in [Section 2.3](#).

Example `<poll>` 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:    <poll op="req"/>
C:      <clTRID>ABC-12345</clTRID>
C:    </command>
C:</epp>
```

Example `<poll>` response with the Registry Maintenance poll message:

```
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="1301">
S:      <msg>Command completed successfully; ack to dequeue</msg>
S:    </result>
S:    <msgQ count="1" id="12345">
S:      <qDate>2020-10-08T22:10:00Z</qDate>
S:      <msg lang="en">Registry Maintenance Notification</msg>
S:    </msgQ>
S:    <resData>
S:      <maint:infData
S:        xmlns:maint="urn:ietf:params:xml:ns:epp:maintenance-0.1">
S:        <maint:item>
S:          <maint:id>2e6df9b0-4092-4491-bcc8-9fb2166dcee6</maint:id>
S:          <maint:systems>
S:            <maint:system>
S:              <maint:name>EPP</maint:name>
S:              <maint:host>epp.registry.example
S:            </maint:host>
S:              <maint:impact>full</maint:impact>
S:            </maint:system>
S:          </maint:systems>
S:        </maint:infData>
S:      </resData>
S:    </response>
S:</epp>
```

S: <maint:environment type="production"/>
S: <maint:start>2020-10-30T06:00:00Z</maint:start>
S: <maint:end>2020-10-30T14:25:57Z</maint:end>
S: <maint:reason>planned</maint:reason>

```
S:      <maint:detail>
S:      https://www.registry.example/notice?123
S:      </maint:detail>
S:      <maint:tlds>
S:      <maint:tld>example</maint:tld>
S:      <maint:tld>test</maint:tld>
S:      </maint:tlds>
S:      <maint:intervention>
S:      <maint:connection>>false</maint:connection>
S:      <maint:implementation>>false</maint:implementation>
S:      </maint:intervention>
S:      <maint:crDate>2020-02-08T22:10:00Z</maint:crDate>
S:      </maint:item>
S:      </maint:infData>
S: </resData>
S: <trID>
S:   <clTRID>ABC-12345</clTRID>
S:   <svTRID>54321-XYZ</svTRID>
S: </trID>
S: </response>
S:</epp>
```

3.2. EPP Transform Commands

EPP provides five commands to transform objects: <create> to create an instance of an object, <delete> to delete an instance of an object, <renew> to extend the validity period of an object, <transfer> to manage object sponsorship changes, and <update> to change information associated with an object.

3.2.1. EPP <create> Command

Create semantics do not apply to maintenance objects, so there is no mapping defined for the EPP <create> command.

3.2.2. EPP <delete> Command

Delete semantics do not apply to maintenance objects, so there is no mapping defined for the EPP <delete> command.

3.2.3. EPP <renew> Command

Renew semantics do not apply to maintenance objects, so there is no mapping defined for the EPP <renew> command.

3.2.4. EPP <transfer> Command

Transfer semantics do not apply to maintenance objects, so there is no mapping defined for the EPP <transfer> command.

3.2.5. EPP <update> Command

Update semantics do not apply to maintenance objects, so there is no mapping defined for the EPP <update> command.

Sattler, et al.

Expires June 30, 2021

[Page 11]

4. Formal Syntax

One schema is presented here that is the EPP Registry Maintenance schema.

The formal syntax presented here is a complete schema representation of the object mapping suitable for automated validation of EPP XML instances. The BEGIN and END tags are not part of the schema; they are used to note the beginning and end of the schema for URI registration purposes.

4.1. Registry Maintenance EPP Mapping Schema

BEGIN

```
<?xml version="1.0" encoding="UTF-8"?>
  <schema targetNamespace="urn:ietf:params:xml:ns:epp:
    maintenance-0.1"
    xmlns:eppcom="urn:ietf:params:xml:ns:eppcom-1.0"
    xmlns:epp="urn:ietf:params:xml:ns:epp-1.0"
    xmlns:maint="urn:ietf:params:xml:ns:epp:maintenance-0.1"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified">

    <!--
    Import common element types
    -->
    <import namespace="urn:ietf:params:xml:ns:eppcom-1.0"/>
    <import namespace="urn:ietf:params:xml:ns:epp-1.0"/>

    <annotation>
      <documentation>
        Extensible Provisioning Protocol v1.0
        Maintenance Mapping Schema.
      </documentation>
    </annotation>

    <!--
    Child elements found in EPP commands.
    -->
    <element name="info" type="maint:infoType"/>

    <!--
    Child elements of the <info> command.
    -->
    <complexType name="infoType">
      <sequence>
        <choice>
          <element name="list"/>
          <element name="id" type="maint:idType"/>
        </choice>
      </sequence>
    </complexType>
  </schema>
END
```

```
</choice>  
</sequence>  
</complexType>
```

Sattler, et al.

Expires June 30, 2021

[Page 12]

```
<!--
Human-readable text may describe the maintenance
-->
<complexType name="idType">
  <simpleContent>
    <extension base="token">
      <attribute name="msg" type="token"/>
      <attribute name="lang" type="language" default="en"/>
    </extension>
  </simpleContent>
</complexType>

<!--
  Info Response element
-->
<element name="infData" type="maint:infDataType"/>

<!--
  <info> response elements.
-->
<complexType name="infDataType">
  <choice>
    <element name="list" type="maint:listDataType"/>
    <element name="item" type="maint:maintDataType"/>
  </choice>
</complexType>

<!--
  Attributes associated with the list info response
-->
<complexType name="listDataType">
  <sequence>
    <element name="listItem" type="maint:maintItemType"
      minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>

<!--
  Attributes associated with the list item info response
-->
<complexType name="maintItemType">
  <sequence>
    <element name="id" type="maint:idType"/>
    <element name="start" type="dateTime" minOccurs="0"/>
    <element name="end" type="dateTime" minOccurs="0"/>
    <element name="crDate" type="dateTime"/>
    <element name="upDate" type="dateTime" minOccurs="0"/>
  </sequence>
</complexType>
```



```
<!--
  Attributes associated with the maintenance info response
-->
<complexType name="maintDataType">
  <sequence>
    <element name="id" type="maint:idType"/>
    <element name="systems" type="maint:systemsType"/>
    <element name="environment" type="maint:envType"/>
    <element name="start" type="dateTime" minOccurs="0"/>
    <element name="end" type="dateTime" minOccurs="0"/>
    <element name="reason" type="maint:reasonEnum"/>
    <element name="detail" type="anyURI" minOccurs="0"/>
    <element name="description" type="maint:descriptionType"
      minOccurs="0"/>
    <element name="tlds" type="maint:tldsType" minOccurs="0"/>
    <element name="intervention" type="maint:interventionType"
      minOccurs="0"/>
    <element name="crDate" type="dateTime"/>
    <element name="upDate" type="dateTime" minOccurs="0"/>
  </sequence>
</complexType>

<!--
  systems element
-->
<complexType name="systemsType">
  <sequence>
    <element name="system" type="maint:systemType"
      maxOccurs="unbounded"/>
  </sequence>
</complexType>

<!--
  Enumerated list of impacts
-->
<simpleType name="impactEnum">
  <restriction base="token">
    <enumeration value="partial"/>
    <enumeration value="full"/>
  </restriction>
</simpleType>

<!--
  description element
-->
<complexType name="descriptionType">
  <simpleContent>
    <extension base="string">
      <attribute name="lang" type="language" default="en"/>
    </extension>
  </simpleContent>
</complexType>
```

```
</extension>  
</simpleContent>  
</complexType>
```

Sattler, et al.

Expires June 30, 2021

[Page 14]

```
<!--
  system element
-->
<complexType name="systemType">
  <sequence>
    <element name="name" type="token"/>
    <element name="host" type="maint:addrType"/>
    <element name="impact" type="maint:impactEnum"/>
  </sequence>
</complexType>

<!--
  host element
-->
<complexType name="addrType">
  <simpleContent>
    <extension base="maint:addrStringType">
      <attribute name="ip" type="maint:ipType" default="v4"/>
    </extension>
  </simpleContent>
</complexType>

<simpleType name="addrStringType">
  <restriction base="token">
    <minLength value="3"/>
    <maxLength value="45"/>
  </restriction>
</simpleType>

<simpleType name="ipType">
  <restriction base="token">
    <enumeration value="v4"/>
    <enumeration value="v6"/>
  </restriction>
</simpleType>

<!--
  Enumerated list of environments
-->
<simpleType name="envEnum">
  <restriction base="token">
    <enumeration value="production"/>
    <enumeration value="ote"/>
    <enumeration value="staging"/>
    <enumeration value="dev"/>
    <enumeration value="custom"/>
  </restriction>
</simpleType>
```



```
<!--
  environment element
-->
<complexType name="envType">
  <simpleContent>
    <extension base="token">
      <attribute name="type" type="maint:envEnum" use="required"/>
      <attribute name="name" type="token" use="optional"/>
    </extension>
  </simpleContent>
</complexType>

<!--
  Enumerated list of reasons
-->
<simpleType name="reasonEnum">
  <restriction base="token">
    <enumeration value="planned"/>
    <enumeration value="emergency"/>
  </restriction>
</simpleType>

<!--
  tlds element
-->
<complexType name="tldsType">
  <sequence>
    <element name="tld" type="eppcom:labelType"
      maxOccurs="unbounded"/>
  </sequence>
</complexType>

<!--
  intervention element
-->
<complexType name="interventionType">
  <sequence>
    <element name="connection" type="boolean"/>
    <element name="implementation" type="boolean"/>
  </sequence>
</complexType>
```

```
<!--  
  End of schema.  
-->  
</schema>  
END
```

5. IANA Considerations

5.1. XML Namespace

This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism defined in [[RFC3688](#)].

Registration request for the maintenance namespace:

URI: urn:ietf:params:xml:ns:epp:maintenance-0.1

Registrant Contact: IESG

XML: None. Namespace URIs do not represent an XML specification.

Registration request for the maintenance schema:

URI: urn:ietf:params:xml:schema:maintenance-1.0

Registrant Contact: IESG

XML: See the "Formal Syntax" section of this document.

5.2. EPP Extension Registry

The following registration of the EPP Extension Registry, described in [[RFC7451](#)], is requested:

Name of Extension: "Registry Maintenance Notifications for the Extensible Provisioning Protocol (EPP)"

Document status: Standards Track

Reference: (insert the reference to RFC version of this document)

Registrant Name and Email Address: IESG, <iesg@ietf.org>

TLDs: Any

IPR Disclosure: None

Status: Active

Notes: None

6. Security Considerations

The mapping extensions described in this document do not provide any security services beyond those specified by EPP [[RFC5730](#)] and protocol layers used by EPP. The security considerations described in these other specifications apply to this specification as well.

7. Implementation Status

Note to RFC Editor: Please remove this section and the reference to [[RFC7942](#)] before publication.

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [[RFC7942](#)]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [[RFC7942](#)], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

7.1. GoDaddy Registry

Organization: GoDaddy Registry

Name: GoDaddy Registry

Description: GoDaddy Registry provides maintenance notifications to their registrars.

Level of maturity: Production

Coverage: All aspects of the protocol according to the draft version 2 are implemented with further updates to come.

Licensing: Proprietary

Contact: quoc@registry.godaddy

URL: <https://registry.godaddy>

Sattler, et al.

Expires June 30, 2021

[Page 18]

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), 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>>.
- [RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, [RFC 5730](#), August 2009, <<https://www.rfc-editor.org/info/rfc5730>>.

8.2. Informative References

- [RFC3339] Klyne, G., Ed. and C. Newman, "Date and Time on the Internet: Timestamps", [RFC 3339](#), July 2002, <<https://www.rfc-editor.org/info/rfc3339>>.
- [RFC5891] Klensin, J., "Internationalized Domain Names in Applications (IDNA): Protocol", [RFC 5891](#), August 2010, <<https://www.rfc-editor.org/info/rfc5891>>.
- [RFC5952] Kawamura, S. and Kawashima, M., "A Recommendation for IPv6 Address Text Representation", [RFC 5952](#), August 2010, <<https://www.rfc-editor.org/info/rfc5952>>.
- [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>>.
- [RFC7942] Sheffer, Y. and Farrel, A., "Improving Awareness of Running Code: The Implementation Status Section", [RFC 7942](#), July 2016, <<https://www.rfc-editor.org/info/rfc7942>>.

Appendix A. Change History

A.1. Change from [draft-sattler-epp-poll-maintenance-response](#) to [draft-sattler-epp-registry-maintenance](#)

Updated to be EPP based instead of JSON document.

A.2. Change from [draft-sattler-epp-registry-maintenance](#) to [draft-ietf-regext-epp-registry-maintenance](#)

Adopted by the REGEXT working group.

[A.3.](#) Change from 00 to 01

Clarified maint:description and maint:environment. Changed maint:description from complexType to simpleType. Fixed typo. Added acknowledgment.

[A.4.](#) Change from 01 to 02

Update language from Domain Name Registry to Registry. Clarified XML namespace urn:ietf:params:xml:ns:maintenance-1.0. Changed host to contain hostName and hostAddr. Changed maint:tlds from MUST to SHOULD. Fixed maint:status in Schema. Changed UUID to a server unique id.

[A.5.](#) Change from 02 to 03

Changed maint:connection from MUST to SHOULD.

[A.6.](#) Change from 03 to 04

A lot of clarifications and editorial changes.

[A.7.](#) Change from 04 to 05

Changed XML namespace from urn:ietf:params:xml:ns:maintenance-1.0 to urn:ietf:params:xml:ns:epp:maintenance-0.1. Removed <maint:status>. Clarified <maint:info> for retrieving maintenance items and the list.

[A.8.](#) Change from 05 to 06

Changed dates in examples to more recent dates. Renamed Query Maintenance Item and List to Info Maintenance Item and List. Removed blackout in favor of full. Added GoDaddy Registry implementation.

Acknowledgments

The authors wish to thank the following individuals for their feedback and suggestions (sorted alphabetically by company):

- o Patrick Mevzek
- o Anthony Eden, DNSimple
- o Christopher Martens, Donuts
- o Quoc-Anh Pham, GoDaddy Registry
- o Raymond Zylstra, GoDaddy Registry
- o Neal McPherson, IONOS
- o Andreas Huber, united-domains
- o Craig Marchant, VentraIP
- o James Gould, Verisign

Authors' Addresses

Tobias Sattler

Email: tobias.sattler@me.com

URI: <https://tobiassattler.com>

Roger Carney

GoDaddy Inc.

14455 N. Hayden Rd. #219

Scottsdale, AZ 85260

US

Email: rcarney@godaddy.com

URI: <http://www.godaddy.com>

Jody Kolker

GoDaddy Inc.

14455 N. Hayden Rd. #219

Scottsdale, AZ 85260

US

Email: jkolker@godaddy.com

URI: <http://www.godaddy.com>