

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 23 August 2021

J. Gould  
VeriSign, Inc.  
M. Casanova  
SWITCH  
19 February 2021

Extensible Provisioning Protocol (EPP) Unhandled Namespaces  
draft-ietf-regext-unhandled-namespaces-08

Abstract

The Extensible Provisioning Protocol (EPP), as defined in RFC 5730, includes a method for the client and server to determine the objects to be managed during a session and the object extensions to be used during a session. The services are identified using namespace URIs, and an "unhandled namespace" is one that is associated with a service not supported by the client. This document defines an operational practice that enables the server to return information associated with unhandled namespace URIs that is compliant with the negotiated services defined in RFC 5730.

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 23 August 2021.

Copyright Notice

Copyright (c) 2021 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.	Conventions Used in This Document . . . . .	3
2.	Unhandled Namespaces . . . . .	4
3.	Use of EPP <extValue> for Unhandled Namespace Data . . . . .	4
3.1.	Unhandled Object-Level Extension . . . . .	5
3.2.	Unhandled Command-Response Extension . . . . .	7
4.	Signaling Client and Server Support . . . . .	10
5.	Usage with General EPP Responses . . . . .	11
6.	Usage with Poll Message EPP Responses . . . . .	13
7.	Implementation Considerations . . . . .	16
7.1.	Client Implementation Considerations . . . . .	16
7.2.	Server Implementation Considerations . . . . .	16
8.	IANA Considerations . . . . .	17
8.1.	XML Namespace . . . . .	17
8.2.	EPP Extension Registry . . . . .	17
9.	Implementation Status . . . . .	17
9.1.	Verisign EPP SDK . . . . .	18
9.2.	SWITCH Automated DNSSEC Provisioning Process . . . . .	18
10.	Security Considerations . . . . .	19
11.	Acknowledgements . . . . .	19
12.	References . . . . .	19
12.1.	Normative References . . . . .	19
12.2.	Informative References . . . . .	20
Appendix A.	Change History . . . . .	20
A.1.	Change from 00 to 01 . . . . .	20
A.2.	Change from 01 to 02 . . . . .	21
A.3.	Change from 02 to REGEXT 00 . . . . .	21
A.4.	Change from REGEXT 00 to REGEXT 01 . . . . .	21
A.5.	Change from REGEXT 01 to REGEXT 02 . . . . .	21
A.6.	Change from REGEXT 02 to REGEXT 03 . . . . .	21
A.7.	Change from REGEXT 03 to REGEXT 04 . . . . .	21
A.8.	Change from REGEXT 04 to REGEXT 05 . . . . .	22
A.9.	Change from REGEXT 05 to REGEXT 06 . . . . .	22
A.10.	Change from REGEXT 06 to REGEXT 07 . . . . .	22
A.11.	Change from REGEXT 07 to REGEXT 08 . . . . .	22
Authors' Addresses	. . . . .	23

## 1. Introduction

The Extensible Provisioning Protocol (EPP), as defined in [RFC5730], includes a method for the client and server to determine the objects to be managed during a session and the object extensions to be used during a session. The services are identified using namespace URIs. How should the server handle service data that needs to be returned in the response when the client does not support the required service namespace URI, which is referred to as an unhandled namespace? An unhandled namespace is a significant issue for the processing of [RFC5730] poll messages, since poll messages are inserted by the server prior to knowing the supported client services, and the client needs to be capable of processing all poll messages. Returning an unhandled namespace poll message is not compliant with the negotiated services defined in [RFC5730] and returning an error makes the unhandled namespace poll message a poison message by halting the processing of the poll queue. An unhandled namespace is an issue also for general EPP responses when the server has information that it cannot return to the client due to the client's supported services. The server should be able to return unhandled namespace information that the client can process later. This document defines an operational practice that enables the server to return information associated with unhandled namespace URIs that is compliant with the negotiated services defined in [RFC5730].

### 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.

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

In examples, "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.

The examples reference XML namespace prefixes that are used for the associated XML namespaces. Implementations MUST NOT depend on the example XML namespaces and instead employ a proper namespace-aware XML parser and serializer to interpret and output the XML documents. The example namespace prefixes used and their associated XML namespaces include:

```
"changePoll": urn:ietf:params:xml:ns:changePoll-1.0
"domain": urn:ietf:params:xml:ns:domain-1.0
"secDNS": urn:ietf:params:xml:ns:secDNS-1.1
```

In the template example XML, placeholder content is represented by the following variables:

```
"[NAMESPACE-XML]": XML content associated with a login service
                    namespace URI. An example is the <domain:infData> element
                    content in [RFC5731].
"[NAMESPACE-URI]": XML namespace URI associated with the [NAMESPACE-
                    XML] XML content. An example is "urn:ietf:params:xml:ns:domain-
                    1.0" in [RFC5731].
```

## 2. Unhandled Namespaces

An Unhandled Namespace is an XML namespace that is associated with a response extension that is not included in the client-specified EPP login services of [RFC5730]. The EPP login services consists of the set of XML namespace URIs included in the <objURI> or <extURI> elements of the [RFC5730] EPP <login> command. The services supported by the server are included in the <objURI> and <extURI> elements of the [RFC5730] EPP <greeting>, which should be a superset of the login services included in the EPP <login> command. A server may have information associated with a specific namespace that it needs to return in the response to a client. The unhandled namespaces problem exists when the server has information that it needs to return to the client but the namespace of the information is not supported by the client based on the negotiated EPP <login> command services.

## 3. Use of EPP <extValue> for Unhandled Namespace Data

In [RFC5730], the <extValue> element is used to provide additional error diagnostic information, including the <value> element that identifies the client-provided element that caused a server error condition and the <reason> element containing the human-readable message that describes the reason for the error. This operational practice extends the use of the <extValue> element for the purpose of returning unhandled namespace information in a successful response.

When a server has data to return to the client that the client does not support based on the login services, the server MAY return a successful response, with the data for each unsupported namespace moved into an [RFC5730] <extValue> element. The unhandled namespace will not cause an error response, but the unhandled namespace data will instead be moved to an <extValue> element, along with a reason why the unhandled namespace data could not be included in the appropriate location of the response. The <extValue> element XML will not be processed by the XML processor. The <extValue> element contains the following child elements:

<value>: Contains a child-element with the unhandled namespace XML. The unhandled namespace MUST be declared in the child element or any containing element including the root element. XML processing of the <value> element is disabled by the XML schema in [RFC5730], so the information can safely be returned in the <value> element.

<reason>: A formatted human-readable message that indicates the reason the unhandled namespace data was not returned in the appropriate location of the response. The formatted reason SHOULD follow the Augmented Backus-Naur Form (ABNF) grammar [RFC5234] format: `NAMESPACE-URI "not in login services"`, where `NAMESPACE-URI` is the unhandled XML namespace like `"urn:ietf:params:xml:ns:domain-1.0"` for [RFC5731].

This document applies to the handling of unsupported namespaces for [RFC3735] object-level extensions and command-response extensions. This document does not apply to the handling of unsupported namespaces for [RFC3735] protocol-level extensions or authentication information extensions. Refer to the following sections on how to handle an unsupported object-level extension namespace or an unsupported command-response extension namespace.

### 3.1. Unhandled Object-Level Extension

An object-level extension in [RFC5730] is a child element of the <resData> element. If the client does not handle the namespace of the object-level extension, then the <resData> element is removed and its object-level extension child element is moved into a [RFC5730] <extValue> <value> element, with the namespace URI included in the corresponding <extValue> <reason> element. The response becomes a general EPP response without the <resData> element.

Template response for a supported object-level extension. The [NAMESPACE-XML] variable represents the object-level extension XML.

```

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:      [NAMESPACE-XML]
S:    </resData>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>

```

Template for an unhandled namespace response for an unsupported object-level extension. The [NAMESPACE-XML] variable represents the object-level extension XML and the [NAMESPACE-URI] variable represents the object-level extension XML namespace URI.

```

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:      <extValue>
S:        <value>
S:          [NAMESPACE-XML]
S:        </value>
S:      <reason>
S:        [NAMESPACE-URI] not in login services
S:      </reason>
S:    </extValue>
S:  </result>
S:  <trID>
S:    <clTRID>ABC-12345</clTRID>
S:    <svTRID>54322-XYZ</svTRID>
S:  </trID>
S: </response>
S:</epp>

```

The EPP response is converted from an object response to a general EPP response by the server when the client does not support the object-level extension namespace URI. Below is an example of converting the <transfer> query response example in Section 3.1.3 of [RFC5731] to an unhandled namespace response.

[RFC5731] example <transfer> query response converted into an unhandled namespace response:

```
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:      <extValue>
S:        <value>
S:          <domain:trnData
S:            xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:              <domain:name>example.com</domain:name>
S:              <domain:trStatus>pending</domain:trStatus>
S:              <domain:reID>ClientX</domain:reID>
S:              <domain:reDate>2000-06-06T22:00:00.0Z</domain:reDate>
S:              <domain:acID>ClientY</domain:acID>
S:              <domain:acDate>2000-06-11T22:00:00.0Z</domain:acDate>
S:              <domain:exDate>2002-09-08T22:00:00.0Z</domain:exDate>
S:            </domain:trnData>
S:          </value>
S:          <reason>
S:            urn:ietf:params:xml:ns:domain-1.0 not in login services
S:          </reason>
S:        </extValue>
S:      </result>
S:      <trID>
S:        <clTRID>ABC-12345</clTRID>
S:        <svTRID>54322-XYZ</svTRID>
S:      </trID>
S:    </response>
S:</epp>
```

### 3.2. Unhandled Command-Response Extension

A command-response extension in [RFC5730] is a child element of the <extension> element. If the client does not handle the namespace of the command-response extension, the command-response child element is moved into an [RFC5730] <extValue> <value> element, with the namespace URI included in the corresponding <extValue> <reason> element. If after moving the command-response child element there are no additional command-response child elements, the <extension> element MUST be removed.

Template response for a supported command-response extension. The [NAMESPACE-XML] variable represents the command-response extension XML.

```

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:    <extension>
S:      [NAMESPACE-XML]
S:    </extension>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>

```

Template unhandled namespace response for an unsupported command-response extension. The [NAMESPACE-XML] variable represents the command-response extension XML and the [NAMESPACE-URI] variable represents the command-response extension XML namespace URI.

```

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:      <extValue>
S:        <value>
S:          [NAMESPACE-XML]
S:        </value>
S:      <reason>
S:        [NAMESPACE-URI] not in login services
S:      </reason>
S:    </extValue>
S:  </result>
S:  <trID>
S:    <clTRID>ABC-12345</clTRID>
S:    <svTRID>54322-XYZ</svTRID>
S:  </trID>
S: </response>
S:</epp>

```

The EPP response is converted to an unhandled namespace response by moving the unhandled command-response extension from under the <extension> to an <extValue> element. Below is example of converting the DS Data Interface <info> response example in Section 5.1.2 of [RFC5910] to an unhandled namespace response.

[RFC5910] DS Data Interface <info> response converted into an unhandled namespace response:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
S:  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:      <extValue>
S:        <value>
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>3</secDNS:alg>
S:                <secDNS:digestType>1</secDNS:digestType>
S:                <secDNS:digest>49FD46E6C4B45C55D4AC</secDNS:digest>
S:              </secDNS:dsData>
S:            </secDNS:infData>
S:          </value>
S:          <reason>
S:            urn:ietf:params:xml:ns:secDNS-1.1 not in login services
S:          </reason>
S:        </extValue>
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:registrant>jd1234</domain:registrant>
S:          <domain:contact type="admin">sh8013</domain:contact>
S:          <domain:contact type="tech">sh8013</domain:contact>
S:          <domain:ns>
S:            <domain:hostObj>ns1.example.com</domain:hostObj>
S:            <domain:hostObj>ns2.example.com</domain:hostObj>
S:          </domain:ns>
S:          <domain:host>ns1.example.com</domain:host>
S:          <domain:host>ns2.example.com</domain:host>
S:          <domain:clID>ClientX</domain:clID>
S:          <domain:crID>ClientY</domain:crID>
S:          <domain:crDate>1999-04-03T22:00:00.0Z</domain:crDate>
S:          <domain:upID>ClientX</domain:upID>
S:          <domain:upDate>1999-12-03T09:00:00.0Z</domain:upDate>
S:          <domain:exDate>2005-04-03T22:00:00.0Z</domain:exDate>
S:          <domain:trDate>2000-04-08T09:00:00.0Z</domain:trDate>
```

```
S:      <domain:authInfo>
S:      <domain:pw>2fooBAR</domain:pw>
S:      </domain:authInfo>
S:      </domain:infData>
S:      </resData>
S:      <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:      </trID>
S:      </response>
S:</epp>
```

#### 4. Signaling Client and Server Support

This document does not define new EPP protocol elements but rather specifies an operational practice using the existing EPP protocol, where the client and the server can signal support for the operational practice using a namespace URI in the login and greeting extension services. The namespace URI "urn:ietf:params:xml:ns:epp:unhandled-namespaces-1.0" is used to signal support for the operational practice. The client includes the namespace URI in an <svcExtension> <extURI> element of the [RFC5730] <login> Command. The server includes the namespace URI in an <svcExtension> <extURI> element of the [RFC5730] Greeting.

A client that receives the namespace URI in the server's Greeting extension services can expect the following supported behavior by the server:

1. Support unhandled namespace object-level extensions and command-response extensions in EPP poll messages, per Section 6.
2. Support the option of unhandled namespace command-response extensions in general EPP responses, per Section 5.

A server that receives the namespace URI in the client's <login> Command extension services can expect the following supported behavior by the client:

1. Support monitoring the EPP poll messages and general EPP responses for unhandled namespaces.

## 5. Usage with General EPP Responses

The unhandled namespace approach defined in Section 3 MAY be used for a general EPP response to an EPP command. A general EPP response includes any non-poll message EPP response. The use of the unhandled namespace approach for poll message EPP responses is defined in Section 6. The server MAY exclude the unhandled namespace information in the general EPP response or MAY include it using the unhandled namespace approach.

The unhandled namespace approach for general EPP responses SHOULD only be applicable to command-response extensions, defined in Section 3.2, since the server SHOULD NOT accept an object-level EPP command if the client did not include the object-level namespace URI in the login services. An object-level EPP response extension is returned when the server successfully executes an object-level EPP command extension. The server MAY return an unhandled object-level extension to the client as defined in Section 3.1.

Returning domain name Redemption Grace Period (RGP) data, based on [RFC3915], provides an example of applying the unhandled namespace approach for a general EPP response. If the client does not include the "urn:ietf:params:xml:ns:rgp-1.0" namespace URI in the login services, and the domain <info> response of a domain name does have RGP information, the server MAY exclude the <rgp:infData> element from the EPP response or MAY include it under the <extValue> element per Section 3.2. Below is example of converting the domain name <info> response example in Section 4.1.2 of [RFC3915] to an unhandled namespace response.

[RFC5731] domain name <info> response with the unhandled [RFC3915] <rgp:infData> element included under an <extValue> element:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
S:   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
S:   xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0
S:     epp-1.0.xsd">
S:   <response>
S:     <result code="1000">
S:       <msg>Command completed successfully</msg>
S:       <extValue>
S:         <value>
S:           <rgp:infData xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
S:             xsi:schemaLocation="urn:ietf:params:xml:ns:rgp-1.0
S:               rgp-1.0.xsd">
S:             <rgp:rgpStatus s="redemptionPeriod"/>
S:           </rgp:infData>
```

```
S:         </value>
S:         <reason>
S:           urn:ietf:params:xml:ns:rgp-1.0 not in login services
S:         </reason>
S:       </extValue>
S:     </result>
S:   <resData>
S:     <domain:infData
S:       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
S:       xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0
S:       domain-1.0.xsd">
S:       <domain:name>example.com</domain:name>
S:       <domain:roid>EXAMPLE1-REP</domain:roid>
S:       <domain:status s="pendingDelete"/>
S:       <domain:registrant>jdl1234</domain:registrant>
S:       <domain:contact type="admin">sh8013</domain:contact>
S:       <domain:contact type="tech">sh8013</domain:contact>
S:       <domain:ns>
S:         <domain:hostObj>ns1.example.com</domain:hostObj>
S:         <domain:hostObj>ns1.example.net</domain:hostObj>
S:       </domain:ns>
S:       <domain:host>ns1.example.com</domain:host>
S:       <domain:host>ns2.example.com</domain:host>
S:       <domain:clID>ClientX</domain:clID>
S:       <domain:crID>ClientY</domain:crID>
S:       <domain:crDate>1999-04-03T22:00:00.0Z</domain:crDate>
S:       <domain:upID>ClientX</domain:upID>
S:       <domain:upDate>1999-12-03T09:00:00.0Z</domain:upDate>
S:       <domain:exDate>2005-04-03T22:00:00.0Z</domain:exDate>
S:       <domain:trDate>2000-04-08T09:00:00.0Z</domain:trDate>
S:       <domain:authInfo>
S:         <domain:pw>2fooBAR</domain:pw>
S:       </domain:authInfo>
S:     </domain:infData>
S:   </resData>
S: <trID>
S:   <clTRID>ABC-12345</clTRID>
S:   <svTRID>54322-XYZ</svTRID>
S: </trID>
S: </response>
S: </epp>
```

## 6. Usage with Poll Message EPP Responses

The unhandled namespace approach, defined in Section 3, MUST be used if there is unhandled namespace information included in an EPP <poll> message response. The server inserts poll messages into the client's poll queue independent of knowing the supported client login services, therefore there may be unhandled object-level and command-response extensions included in a client's poll queue. In [RFC5730], the <poll> command is used by the client to retrieve and acknowledge poll messages that have been inserted by the server. The <poll> message response is an EPP response that includes the <msgQ> element that provides poll queue meta-data about the message. The unhandled namespace approach, defined in Section 3, is used for an unhandled object-level extension and for each of the unhandled command-response extensions attached to the <poll> message response. The resulting EPP <poll> message response MAY have either or both the object-level extension or command-response extensions moved to <extValue> elements, as defined in Section 3.

The Change Poll Message, as defined in Section 3.1.2 of [RFC8590], which is an extension of any EPP object, is an example of applying the unhandled namespace approach for EPP <poll> message responses. Below are examples of converting the domain name <info> response example in Section 3.1.2 of [RFC8590] to an unhandled namespace response. The object that will be used in the examples is a [RFC5731] domain name object.

[RFC5731] domain name <info> <poll> message response with the unhandled [RFC8590] <changePoll:changeData> element included under an <extValue> element:

```
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 lang="en-US">
S:        Command completed successfully; ack to dequeue</msg>
S:      <extValue>
S:        <value>
S:          <changePoll:changeData
S:            xmlns:changePoll="urn:ietf:params:xml:ns:changePoll-1.0"
S:            state="after">
S:            <changePoll:operation>update</changePoll:operation>
S:            <changePoll:date>
S:              2013-10-22T14:25:57.0Z</changePoll:date>
S:            <changePoll:svTRID>12345-XYZ</changePoll:svTRID>
S:            <changePoll:who>URS Admin</changePoll:who>
S:            <changePoll:caseId type="urs">urs123
```

```

S:         </changePoll:caseId>
S:         <changePoll:reason>URS Lock</changePoll:reason>
S:         </changePoll:changeData>
S:         </value>
S:         <reason>
S:         urn:ietf:params:xml:ns:changePoll-1.0 not in login services
S:         </reason>
S:         </extValue>
S:     </result>
S:     <msgQ count="201" id="1">
S:         <qDate>2013-10-22T14:25:57.0Z</qDate>
S:         <msg>Registry initiated update of domain.</msg>
S:     </msgQ>
S:     <resData>
S:         <domain:infData
S:             xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:             <domain:name>domain.example</domain:name>
S:             <domain:roid>EXAMPLE1-REP</domain:roid>
S:             <domain:status s="ok"/>
S:             <domain:registrant>jd1234</domain:registrant>
S:             <domain:contact type="admin">sh8013</domain:contact>
S:             <domain:contact type="tech">sh8013</domain:contact>
S:             <domain:clID>ClientX</domain:clID>
S:             <domain:crID>ClientY</domain:crID>
S:             <domain:crDate>2012-04-03T22:00:00.0Z</domain:crDate>
S:             <domain:exDate>2014-04-03T22:00:00.0Z</domain:exDate>
S:         </domain:infData>
S:     </resData>
S:     <trID>
S:         <clTRID>ABC-12345</clTRID>
S:         <svTRID>54322-XYZ</svTRID>
S:     </trID>
S: </response>
S:</epp>

```

Unhandled [RFC5731] domain name <info> <poll> message response and the unhandled [RFC8590] <changePoll:changeData> element included under an <extValue> element:

```

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:      <extValue>
S:        <value>
S:          <domain:infData
S:            xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">

```

```

S:         <domain:name>domain.example</domain:name>
S:         <domain:roid>EXAMPLE1-REP</domain:roid>
S:         <domain:status s="ok"/>
S:         <domain:registrant>jdl234</domain:registrant>
S:         <domain:contact type="admin">sh8013</domain:contact>
S:         <domain:contact type="tech">sh8013</domain:contact>
S:         <domain:clID>ClientX</domain:clID>
S:         <domain:crID>ClientY</domain:crID>
S:         <domain:crDate>2012-04-03T22:00:00.0Z</domain:crDate>
S:         <domain:exDate>2014-04-03T22:00:00.0Z</domain:exDate>
S:         </domain:infData>
S:     </value>
S:     <reason>
S:         urn:ietf:params:xml:ns:domain-1.0 not in login services
S:     </reason>
S: </extValue>
S: <extValue>
S:     <value>
S:         <changePoll:changeData
S:             xmlns:changePoll=
S:                 "urn:ietf:params:xml:ns:changePoll-1.0"
S:             state="after">
S:         <changePoll:operation>update</changePoll:operation>
S:         <changePoll:date>
S:             2013-10-22T14:25:57.0Z</changePoll:date>
S:         <changePoll:svTRID>12345-XYZ</changePoll:svTRID>
S:         <changePoll:who>URS Admin</changePoll:who>
S:         <changePoll:caseId type="urs">urs123
S:         </changePoll:caseId>
S:         <changePoll:reason>URS Lock</changePoll:reason>
S:     </changePoll:changeData>
S: </value>
S: <reason>
S:     urn:ietf:params:xml:ns:changePoll-1.0 not in login services
S: </reason>
S: </extValue>
S: </result>
S: <msgQ count="201" id="1">
S:     <qDate>2013-10-22T14:25:57.0Z</qDate>
S:     <msg>Registry initiated update of domain.</msg>
S: </msgQ>
S: <trID>
S:     <clTRID>ABC-12345</clTRID>
S:     <svTRID>54322-XYZ</svTRID>
S: </trID>
S: </response>
S: </epp>

```

## 7. Implementation Considerations

There are implementation considerations for the client and the server to help address the risk of the client ignoring unhandled namespace information included in an EPP response that is needed to meet technical, policy, or legal requirements.

### 7.1. Client Implementation Considerations

To reduce the likelihood of a client receiving unhandled namespace information, the client should consider implementing the following:

1. Ensure that the client presents the complete set of what it supports when presenting its login services. If there are gaps between the services supported by the client and the login services included in the login command, the client may receive unhandled namespace information that the client could have supported.
2. Support all of the services included in the server greeting services that may be included in an EPP response, including the poll queue responses. The client should evaluate the gaps between the greeting services and the login services provided in the login command to identify extensions that need to be supported.
3. Proactively monitor for unhandled namespace information in the EPP responses by looking for the inclusion of the <extValue> element in successful responses, recording the unsupported namespace included in the <reason> element, and recording the unhandled namespace information included in the <value> element for later processing. The unhandled namespace should be implemented by the client to ensure that information is processed fully in future EPP responses.

### 7.2. Server Implementation Considerations

To assist the clients in recognizing unhandled namespaces, the server should consider implementing the following:

1. Monitor for returning unhandled namespace information to clients and report it to the clients out-of-band to EPP so the clients can add support for the unhandled namespaces.
2. Look for the unhandled namespace support in the login services when returning optional unhandled namespace information in General EPP Responses.

## 8. IANA Considerations

### 8.1. XML Namespace

This document uses URNs to describe XML namespaces conforming to a registry mechanism described in [RFC3688]. The following URI assignment is requested of IANA:

Registration request for the unhandled namespaces namespace:

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

### 8.2. EPP Extension Registry

The EPP operational practice described in this document should be registered by the IANA in the EPP Extension Registry described in [RFC7451]. The details of the registration are as follows:

Name of Extension: "Extensible Provisioning Protocol (EPP) Unhandled Namespaces"

Document status: Standards Track

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

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

TLDs: Any

IPR Disclosure: None

Status: Active

Notes: None

## 9. Implementation Status

Note to RFC Editor: Please remove this section and the reference to RFC 7942 [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 RFC 7942 [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 RFC 7942 [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".

#### 9.1. Verisign EPP SDK

Organization: Verisign Inc.

Name: Verisign EPP SDK

Description: The Verisign EPP SDK includes an implementation of the unhandled namespaces for the processing of the poll queue messages.

Level of maturity: Development

Coverage: All aspects of the protocol are implemented.

Licensing: GNU Lesser General Public License

Contact: [jgould@verisign.com](mailto:jgould@verisign.com)

URL: [https://www.verisign.com/en\\_US/channel-resources/domain-registry-products/epp-sdks](https://www.verisign.com/en_US/channel-resources/domain-registry-products/epp-sdks)

#### 9.2. SWITCH Automated DNSSEC Provisioning Process

Organization: SWITCH

Name: Registry of .CH and .LI

Description: SWITCH uses poll messages to inform the registrar about DNSSEC changes at the registry triggered by CDS records. These poll messages are enriched with the 'urn:ietf:params:xml:ns:changePoll-1.0' and the 'urn:ietf:params:xml:ns:secDNS-1.1' extension that are rendered in the poll msg response according to this draft.

Level of maturity: Operational

Coverage: All aspects of the protocol are implemented.

Licensing: Proprietary

Contact: martin.casanova@switch.ch

URL: <https://www.nic.ch/cds>

## 10. Security Considerations

This document does not provide any security services beyond those described by EPP [RFC5730] and protocol layers used by EPP. The security considerations described in these other specifications apply to this specification as well. Since the unhandled namespace context is XML that is not processed in the first pass by the XML parser, the client SHOULD validate the XML when the content is processed to protect against the inclusion of malicious content.

## 11. Acknowledgements

The authors wish to thank the following persons for their feedback and suggestions: Thomas Corte, Scott Hollenbeck, Patrick Mevzek, and Marcel Parodi.

## 12. References

### 12.1. Normative References

- [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>>.
- [RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <<https://www.rfc-editor.org/info/rfc5234>>.
- [RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, DOI 10.17487/RFC5730, August 2009, <<https://www.rfc-editor.org/info/rfc5730>>.

- [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>>.
- [RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <<https://www.rfc-editor.org/info/rfc7942>>.
- [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>>.

## 12.2. Informative References

- [RFC3735] Hollenbeck, S., "Guidelines for Extending the Extensible Provisioning Protocol (EPP)", RFC 3735, DOI 10.17487/RFC3735, March 2004, <<https://www.rfc-editor.org/info/rfc3735>>.
- [RFC3915] Hollenbeck, S., "Domain Registry Grace Period Mapping for the Extensible Provisioning Protocol (EPP)", RFC 3915, DOI 10.17487/RFC3915, September 2004, <<https://www.rfc-editor.org/info/rfc3915>>.
- [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>>.
- [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>>.

## Appendix A. Change History

### A.1. Change from 00 to 01

1. Removed `xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"` reference from examples.
2. removed `<extension></extension>` block from example.

3. added SWITCH Automated DNSSEC Provisioning Process at Implementation Status
- A.2. Change from 01 to 02
    1. Ping update
  - A.3. Change from 02 to REGEXT 00
    1. Changed to regext working group draft by changing draft-gould-casanova-regext-unhandled-namespaces to draft-ietf-regext-unhandled-namespaces.
  - A.4. Change from REGEXT 00 to REGEXT 01
    1. Added the "Signaling Client and Server Support" section to describe the mechanism to signal support for the BCP by the client and the server.
    2. Added the IANA Considerations section with the registration of the unhandled namespaces XML namespace and the registration of the EPP Best Current Practice (BCP) in the EPP Extension Registry.
  - A.5. Change from REGEXT 01 to REGEXT 02
    1. Filled in the acknowledgements section.
    2. Changed the reference from RFC 5730 to RFC 5731 for the transfer example in section 3.1 "Unhandled Object-Level" Extension.
    3. Updated the XML namespace to urn:ietf:params:xml:ns:epp:unhandled-namespaces-1.0, which removed bcp from the namespace and bumped the version from 0.1 and 1.0. Inclusion of bcp in the XML namespace was discussed at the REGEXT interim meeting.
  - A.6. Change from REGEXT 02 to REGEXT 03
    1. Converted from xml2rfc v2 to v3.
    2. Updated Acknowledgements to match the approach taken by the RFC Editor with draft-ietf-regext-login-security.
    3. Changed reference of ietf-regext-change-poll to RFC 8590.
  - A.7. Change from REGEXT 03 to REGEXT 04
    1. Changed from Best Current Practice (BCP) to Standards Track based on mailing list discussion.
    2. Revised the dates in the examples to be more up-to-date.

## A.8. Change from REGEXT 04 to REGEXT 05

1. Based on feedback from Thomas Corte, added a description of the <extValue> element in RFC 5730 and it being extended to support returning unhandled namespace information.
2. Based on feedback from Thomas Corte, added a Implementation Considerations section to cover client and server implementation recommendations such as monitoring unhandled namespaces in the server to report to the clients out-of-band and monitoring for responses containing unhandled namespace information in the client to proactively add support for the unhandled namespaces.
3. Moved RFC 3735 and RFC 7451 to informative references to address down reference errors in idnits.

## A.9. Change from REGEXT 05 to REGEXT 06

1. Nit updates made based on the feedback provided by the Document Shepherd, David Smith.

## A.10. Change from REGEXT 06 to REGEXT 07

Updates based on the Barry Leiba (AD) feedback:

1. Simplified the abstract based on the proposal provided by the AD.
2. In section 1.1, updated to use the new BCP 14 boilerplate and add a normative reference to RFC 8174.
3. In section 1.1, changed "REQUIRED feature of this protocol" to "required feature of this protocol".
4. In section 3, added "by the XML schema" in "disabled by the XML schema in [RFC5730]" to clarify the statement.
5. In section 8.2, changed the Registrant Name from "IESG" to "IETF".
6. In section 10, changed "The document do not provide" to "This document does not provide".
7. In section 10, added the sentence "Since the unhandled namespace context is XML that is not processed in the first pass by the XML parser, the client SHOULD consider validating the XML when the content is processed to protect against the inclusion of malicious content.".

## A.11. Change from REGEXT 07 to REGEXT 08

1. Nit updates made based on the feedback provided by Peter Yee.
2. Update to the definition of the <value> element based on feedback from Sabrina Tanamal.
3. Added a sentence in the Introduction section to cover the poison poll message motivation based on feedback from Qin Wu.

4. Changed "does not define new protocol" to "does not define new EPP protocol elements" based on feedback from Erik Kline.
5. Changed to use "apply" instead of "support" language in Section 3 based on feedback from Benjamin Kaduk.
6. Updated the examples that reference RFC examples to reference the RFC section of the example and have the starting XML match based on feedback from Benjamin Kaduk.
7. Changed "SHOULD consider validating" to "SHOULD validate" in the Security Considerations section based on feedback from Benjamin Kaduk.
8. Moved RFC 3915, RFC 5910, and RFC 8590 as informational references based on feedback from Benjamin Kaduk.

#### Authors' Addresses

James Gould  
VeriSign, Inc.  
12061 Bluemont Way  
Reston, VA 20190  
United States of America

Email: [jgould@verisign.com](mailto:jgould@verisign.com)  
URI: <http://www.verisigninc.com>

Martin Casanova  
SWITCH  
P.O. Box  
CH-8021 Zurich  
Switzerland

Email: [martin.casanova@switch.ch](mailto:martin.casanova@switch.ch)  
URI: <http://www.switch.ch>