

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 26, 2015

A. Mayrhofer
nic.at
October 23, 2014

EPP Service Messages Extension
draft-mayrhofer-epext-servicemessage-00

Abstract

EPP contains a mechanism to convey service messages to clients via the "poll" query command. This document specifies an extension defining a simple structured format for including additional information in service messages.

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 <http://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 April 26, 2015.

Copyright Notice

Copyright (c) 2014 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 (<http://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	2
2.	Terminology	3
3.	EPP Command Mapping	3
3.1.	EPP Query Commands	3
3.1.1.	EPP <poll> Command	3
3.1.2.	Other Query Commands	4
3.2.	EPP Transform Commands	4
3.3.	Examples	4
3.4.	Formal Syntax	8
4.	Security Considerations	10
5.	IANA Considerations	10
6.	Acknowledgements	11
7.	References	11
7.1.	Normative References	11
7.2.	Informative References	11
	Author's Address	12

[1.](#) Introduction

The Extensible Provisioning Protocol (EPP) [[RFC5730](#)] specifies a queue-based mechanism to convey service messages to clients. Clients can discover, retrieve and acknowledge such service messages via the "poll" query command. This mechanism is widely used in deployments of EPP servers.

The contents of service messages is extensible. While EPP Object Mappings define object-specific messages for some specific situations (For example, see [Section 3.3 of \[RFC5732\]](#)), there is no generic structure defined for the many other situations in which domain name registries require conveying additional information. Such situations include:

- o Notifying a registrar if (and which) additional objects were transferred automatically together with a domain name.
- o Sending warnings to registrars when their credit level reaches a low water mark.
- o Providing a list of domain names that were auto-renewed by the registry on behalf of the client.
- o Sending warning messages to clients when objects in the registry are about to expire (or have expired recently).

As outlined above, information to be sent in such "additional" service messages is typically (although not always) structured. For

interoperability reasons, while it is highly desirable to keep a certain level of structure when conveying this information to clients, a proliferation of custom extensions for each and every situation should be avoided.

This document specifies such a structure as an extension to the "resData" element of EPP service messages. The specification follows the guidelines contained in [[RFC3735](#)].

2. Terminology

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](#)].

3. EPP Command Mapping

3.1. EPP Query Commands

3.1.1. EPP <poll> Command

This extension adds elements to the response to a <poll> command with the "op" attribute set to "req". Specifically, a "message" element is added to the "resData" section of the service message, containing the following elements/attributes:

- o A REQUIRED "type" attribute that identifies the type of service message. The definition of values for this attribute is subject to local server policy.
- o A REQUIRED "desc" element that contains free form text.
- o An OPTIONAL "reftrID" element ("reference transaction identifier") containing:
 - * An OPTIONAL "clTRID" element that contains the client transaction identifier of the EPP transaction that instigated the service message.
 - * A REQUIRED "svTRID" element that contains the server transaction identifier of the EPP transaction that instigated the service message.
- o An OPTIONAL "data" element, containing at least one of:
 - * Zero or more OPTIONAL "entry" elements. Each "entry" element contains a REQUIRED "name" attribute. These elements transport

key/value tuples. The definition of values for the "name" attribute is subject to local server policy.

- * An OPTIONAL "request" element, containing an EPP request frame related to this service message.
- * An OPTIONAL "response" element, containing an EPP response frame related to this service message. Note that if both "request" and "response" elements are used, their contents MUST be from the same EPP transaction.
- * An OPTIONAL "any other" element from "any other" namespace - Note that this is used for legacy reasons in some situations to include EPP frames outside of the request/response elements pair, or other data.

This extension does not add any elements to the <poll> command nor to the response of the <poll> command with the "op" attribute set to "ack".

3.1.2. Other Query Commands

This extension does not add any elements to the <check>, <info> and <transfer> query commands.

3.2. EPP Transform Commands

This extension does not add any elements to any of the EPP transform commands (<create>, <delete>, <renew>, <transfer>, <update>).

3.3. Examples

The following example informs a client that an inbound transfer was approved, and two subordinate host objects were automatically transferred together with the domain name:


```
<epp xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
<response>
  <result code="1301">
    <msg lang="en-US">Command completed successfully; ack to dequeue</msg>
  </result>
  <msgQ count="1" id="137526">
    <qDate>2013-11-27T04:04:51.0Z</qDate>
    <msg lang="en-US">Transfer Approved.</msg>
  </msgQ>
  <resData>
    <domain:trnData xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0
domain-1.0.xsd">
      <domain:name>test.example</domain:name>
      <domain:trStatus>clientApproved</domain:trStatus>
      <domain:reID>reg1</domain:reID>
      <domain:reDate>2013-11-27T04:03:29.0Z</domain:reDate>
      <domain:acID>reg2</domain:acID>
      <domain:acDate>2013-11-27T04:04:51.0Z</domain:acDate>
    </domain:trnData>
    <message xmlns="http://tld-box.at/xmlns/resdata-1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
type="TransferApproved">
      <desc>Inbound transfer of test.example was APPROVED.
Subordinate hosts ns1.test.example, ns2.test.example were also
transferred.</desc>
      <data>
        <entry name="host">ns1.test.example</entry>
        <entry name="host">ns2.test.example</entry>
      </data>
    </message>
  </resData>
  <trID>
    <svTRID>123</svTRID>
  </trID>
</response>
</epp>
```

In this example, the extension is used to inform a client that some domain names have expired recently:


```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">
  <response>
    <result code="1301">
      <msg>Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ count="1" id="2267">
      <qDate>2016-02-25T13:46:36.879301Z</qDate>
      <msg>The following domains have expired as of 2016-02-25:
      test-expire1.example, test-expire2.example</msg>
    </msgQ>
    <resData>
      <message xmlns="http://tld-box.at/xmlns/resdata-1.1" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" type="HasExpired">
        <desc>The following domains have expired as of 2016-02-25:
        test-expire1.example, test-expire2.example</desc>
        <data>
          <entry name="date">2016-02-25</entry>
          <entry name="domain">test-expire1.example</entry>
          <entry name="domain">test-expire2.example</entry>
        </data>
      </message>
    </resData>
    <trID>
      <clTRID>AD59FECE-5928-11E4-8467-BBC5AB10F032</clTRID>
      <svTRID>20141021134636989450F6-primary-tldbox</svTRID>
    </trID>
  </response>
</epp>
```

An interesting use case of this extension is when (for whatever reason) the connection between an EPP server and an EPP client fails while a request is being processed. Subsequently, the server cannot provide the client with the response frame. The following example illustrates how the response frame can be submitted to the client using this extension:


```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">
  <response>
    <result code="1301">
      <msg>Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ count="88" id="1816">
      <qDate>2014-10-21T14:31:54.524131Z</qDate>
      <msg>EPP response to command with client-id [05908A94-592F-11E4-
ABEA-51CFAB10F032]
      and server-id [20141021143201978589AD-secondary-tldbox]</msg>
    </msgQ>
    <resData>
      <message xmlns="http://tld-box.at/xmlns/resdata-1.1" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" type="ResponseRecovery">
        <desc>EPP response to command with client-id [05908A94-592F-11E4-
ABEA-51CFAB10F032]
        and server-id [20141021143201978589AD-secondary-tldbox]</desc>
        <data>
          <epp xmlns="urn:ietf:params:xml:ns:epp-1.0" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance">
            <response>
              <result code="1000">
                <msg>Command completed successfully</msg>
              </result>
              <msgQ count="8" id="1975"/>
              <resData>
                <domain:creData
xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
xmlns="urn:ietf:params:xml:ns:domain-1.0">
                  <domain:name>test-connection-interrupt.example</domain:name>
                  <domain:crDate>2014-10-21T14:32:01.984360Z</domain:crDate>
                  <domain:exDate>2015-10-21T14:32:01.984360Z</domain:exDate>
                </domain:creData>
              </resData>
              <trID>
                <clTRID>05908A94-592F-11E4-ABEA-51CFAB10F032</clTRID>
                <svTRID>20141021143201978589AD-secondary-tldbox</svTRID>
              </trID>
            </response>
          </epp>
        </data>
      </message>
    </resData>
    <trID>
      <clTRID>06706F24-592F-11E4-ABEA-51CFAB10F032</clTRID>
      <svTRID>20141021143203441442CD-primary-tldbox</svTRID>
    </trID>
  </response>
</epp>

```

```
</trID>  
</response>  
</epp>
```

Note that, in the example above, the EPP response frame is included as a child element of the "data" element directly, using the "any"

element of the Schema. This is for legacy reasons of the current implementation, since the Schema described also allows an explicit "response" element that should be preferred for including EPP response frames.

The example might be updated as the implementation is adapted.

The following example informs a client that - for reasons that are outside of the scope of this document - status values on one of the domain names under his sponsorship have been set:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">
  <response>
    <result code="1301">
      <msg>Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ count="1" id="16031">
      <qDate>2014-12-28T13:48:22.097813Z</qDate>
      <msg>Status(es) added to domain [test---0039888rbx-vvgobook5x14.tldbox]:
serverUpdateProhibited (testcase comment freeze (2014-12-28T13:48:22.097813Z)),
serverTransferProhibited (testcase comment freeze
(2014-12-28T13:48:22.097813Z))</msg>
    </msgQ>
    <resData>
      <message xmlns="http://tld-box.at/xmlns/resdata-1.1" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" type="DelegationStatusSet">
        <desc>Status(es) added to domain [test-freeze.example]:
serverUpdateProhibited (testcase comment freeze (2014-12-28T13:48:22.097813Z)),
serverTransferProhibited (testcase comment freeze
(2014-12-28T13:48:22.097813Z))</desc>
        <data>
          <entry name="domain">test-freeze.example</entry>
          <entry name="status">serverUpdateProhibited</entry>
          <entry name="comment">testcase comment freeze
(2014-12-28T13:48:22.097813Z)</entry>
          <entry name="status">serverTransferProhibited</entry>
          <entry name="comment">testcase comment freeze
(2014-12-28T13:48:22.097813Z)</entry>
        </data>
      </message>
    </resData>
    <trID>
      <clTRID>40FD1B64-5ABB-11E4-BFE1-587CAB10F032</clTRID>
      <svTRID>2014102313482238365346-primary-tldbox</svTRID>
    </trID>
```

```
</response>  
</epp>
```

3.4. Formal Syntax

The XML Schema of the extension is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<schema targetNamespace="http://tld-box.at/xmlns/resdata-1.1"
  xmlns:tld="http://tld-box.at/xmlns/resdata-1.1"
  xmlns:epp="urn:ietf:params:xml:ns:epp-1.0"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

  <!-- Import common element types. -->

  <import namespace="urn:ietf:params:xml:ns:epp-1.0"
    schemaLocation="epp-1.0.xsd"/>

  <annotation>
    <documentation>
      Extensible Provisioning Protocol v1.0
      nic.at Service Message extension
draft-mayrhofer-eppext-servicemessage-00
    </documentation>
  </annotation>

  <!-- service message -->

  <element name="message" type="tld:messageType"/>

  <complexType name="messageType">
    <sequence>
      <element name="desc" type="string"/>
      <element name="reftrID" type="tld:reftrIDType"
        minOccurs="0"/>
      <element name="data" type="tld:messageDataType"
        minOccurs="0"/>
    </sequence>
    <attribute name="type" type="tld:messageTypeType"
      use="required"/>
  </complexType>

  <complexType name="frameType">
    <sequence>
      <any namespace="##other"/>
    </sequence>
  </complexType>

  <complexType name="messageDataType">
    <sequence>
      <element name="entry" type="tld:attributeType"
        maxOccurs="unbounded" minOccurs="0"/>
      <element name="request" type="tld:frameType" minOccurs="0"/>
      <element name="response" type="tld:frameType" minOccurs="0"/>
      <any namespace="##other" minOccurs="0"/>
    </sequence>
  </complexType>
```



```
    </sequence>
  </complexType>

  <simpleType name="messageTypeType">
    <restriction base="token">
      <maxLength value="100"/>
    </restriction>
  </simpleType>

  <complexType name="reftrIDType">
    <sequence>
      <element name="clTRID" type="epp:trIDStringType"
        minOccurs="0"/>
      <element name="svTRID" type="epp:trIDStringType"/>
    </sequence>
  </complexType>

  <!-- attributes -->

  <complexType name="attributeType">
    <simpleContent>
      <extension base="string">
        <attribute name="name" type="token"
          use="required"/>
      </extension>
    </simpleContent>
  </complexType>

</schema>
```

4. Security Considerations

This extension allows for inclusion of arbitrary EPP frames in service messages. Since EPP frames can include sensitive information, implementations MUST carefully consider risks related to the use of that mechanism, especially whenever EPP frames from other clients are involved. For example, a "login" frame typically contains authentication credentials, and hence MUST NOT be exposed to a third party client.

5. IANA Considerations

IANA is requested to register the specified extension with the "Extensions for the Extensible Provisioning Protocol Registry" ([\[draft-ietf-eppext-reg-09.txt\]](#)). The required registration information is contained below:

-----BEGIN FORM-----

Name of Extension:

"EPP Service Messages Extension"

Document Status:

Informational

Reference:

<http://tools.ietf.org/html/draft-mayrhofer-eppext-servicemessage-01>

Registrant Name and Email Address:

Alexander Mayrhofer, alexander.mayrhofer@nic.at

TLDs: .at, .berlin, .brussels, .hamburg, .reise, .tirol,
.versicherung, .vlaanderen, .voting, .wien

IPR Disclosure: None

Status: Active

Notes: The .at TLD currently uses the schema under a different
namespace identifier

-----END FORM-----

6. Acknowledgements

The extension was specified and implemented by Achim Adam, Bernhard Reutner-Fischer, Marcel Gruenauer and Mark Hofstetter. Achim Adam also performed reviews, suggested text, and provided the examples contained in this document.

7. References

7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, [RFC 5730](#), August 2009.

7.2. Informative References

- [RFC3735] Hollenbeck, S., "Guidelines for Extending the Extensible Provisioning Protocol (EPP)", [RFC 3735](#), March 2004.

[RFC5732] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)
Host Mapping", STD 69, [RFC 5732](#), August 2009.

Author's Address

A. Mayrhofer
nic.at GmbH
Jakob-Haringer-Strasse 8
Salzburg 5020
AT

Email: alexander.mayrhofer@nic.at