Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 14, 2021

D. Belyavskiy

October 11, 2020

Use of Internationalized Email Addresses in EPP protocol
draft-belyavskiy-epp-eai-01

Abstract

This document permits usage of Internationalized Email Addresses in the EPP protocol. The Extensible Provisioning Protocol (EPP), being developed before appearing the standards for Internationalized Email Addresses (EAI), does not support such email addressed. This document describes an EPP extension that allows EAI addresses to be used in contact mapping in the EPP protocol.

TO BE REMOVED on turning to RFC: The document is edited in the dedicated github repo [1]. Please send your submissions via GitHub.

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 April 14, 2021.

Table of Contents

1. Introduction

RFC 6530 [RFC6530] introduced the framework for Internationalized Email Addresses. To make such addresses more widely accepted, the changes to various protocols need to be introduced.

This document describes an Extensible Provisioning Protocol (EPP) extension for using the Email Addresses Internationalized (EAI) in contact object mapping descibed in RFC 5733 [RFC5733].

The Extensible Provisioning Protocol (EPP) specified in RFC 5730 [RFC5730] is a base document for object management operations and an extensible framework that maps protocol operations to objects. The

specifics of various objects managed via EPP is described in separate documents, RFC 5731 [RFC5731], RFC 5732 [RFC5732], RFC 5733 [RFC5733], and RFC 5734 [RFC5734].

RFC 5733 [RFC5733] describes an Extensible Provisioning Protocol (EPP) mapping for the provisioning and management of individual or organizational social information identifiers (known as "contacts") stored in a shared central repository. This document is the only one refering to Email address as a property of a managed object.

RFC 3735 [RFC3735] provides a guideline to extend the EPP protocol for various purposes.

1.1. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

In examples, "C:" represents lines sent by a protocol client and "S:" represents lines returned by a protocol server. In examples, indentation and whitespace are provided only to illustrate element relationships and are not a required feature of this protocol.

"eppEAI-1.0" is used as an abbreviation for "urn:ietf:params:xml:ns:epp:eppEAI-1.0". The XML namespace prefix "eppEAI" is used, but implementations MUST NOT depend on it. Instead, they are to employ a proper namespace-aware XML parser and serializer to interpret and output the XML documents.

2. Migrating to Newer Versions of This Extension

Servers that implement this extension SHOULD provide a way for clients to progressively update their implementations when a new version of the extension is deployed. A newer version of the extension is expected to use an XML namespace with a higher version number than the prior versions.

3. Object Attributes

This extension adds additional elements to RFC 5733 commands and adds a predefined value to indicate usage of the EAI in the object data. Only those new elements and values are described here.

3.1. EAI email address

An address used by <eppEAI:email> element represents an email address matching the specification in RFC 6530 [RFC6530].

Example EAI email address:


```
<eppEAI:email>
someaddress@example.com
</eppEAI:email>
```

3.2. [EAI-DUMMY] email value

When the RFC 5733 <email> element contains the predefined value of [EAI-DUMMY], the <eppEAI:email> extension MUST be present. The value in the <eppEAI:email> MUST be used as the contact object email attribute.

4. Email Address specification

Email address syntax is defined in in RFC 6530 [RFC6530]. This mapping does not prescribe minimum or maximum lengths for character strings used to represent email addresses.

5. EPP commands mapping

The proposed extension modifies mapping for the <contact:create>, <contact:update>, and <contact:info> commands.

5.1. EPP <create> command

Example <create> command:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <create>
C:      <contact:create
C:       xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
C:        <contact:id>sh8013</contact:id>
C:        <contact:postalInfo type="int">
C:          <contact:name>John Doe</contact:name>
C:          <contact:org>Example Inc.</contact:org>
C:          <contact:addr>
C:            <contact:street>123 Example Dr.</contact:street>
C:            <contact:street>Suite 100</contact:street>
C:            <contact:city>Dulles</contact:city>
C:            <contact:sp>VA</contact:sp>
C:            <contact:pc>20166-6503</contact:pc>
C:            <contact:cc>US</contact:cc>
C:          </contact:addr>
C:        </contact:postalInfo>
C:        <contact:voice x="1234">+1.7035555555</contact:voice>
C:        <contact:fax>+1.7035555556</contact:fax>
C:        <contact:email>[EAI-DUMMY]</contact:email>
```

```
C:        <contact:authInfo>
C:          <contact:pw>2fooBAR</contact:pw>
C:        </contact:authInfo>
C:        <contact:disclose flag="0">
C:          <contact:voice/>
C:          <contact:email/>
C:        </contact:disclose>
C:      </contact:create>
C:    <extension>
C:      <eppEAI:eppEAI
C:        xmlns:eppEAI=
C:          "urn:ietf:params:xml:ns:epp:eppEAI-1.0">
C:        <eppEAI:email>someaddress@example.com</eppEAI:email>
C:      </eppEAI:eppEAI>
C:    </extension>
C:    </create>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

5.2. EPP <update> command

When executing the <update> command, there are multiple possibilities of changing the email address.

No mentioning of <email> element means the email address of the contact is not changed. The <eppEAI> extension MUST not be present.

When the [EAI-DUMMY] of <email> is used, the <eppEAI> extension MUST be present and contain a valid email address.

When the value not equal to [EAI-DUMMY] of <email> is used, the <eppEAI> extension MUST NOT be present.

Example <update> command:

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

```
C:    <update>
C:      <contact:update
C:       xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
C:        <contact:id>sh8013</contact:id>
C:        <contact:email>[EAI-DUMMY]</contact:email>
C:      </contact:update>
C:    <extension>
C:      <eppEAI:eppEAI
C:        xmlns:eppEAI=
C:          "urn:ietf:params:xml:ns:epp:eppEAI-1.0">
C:        <eppEAI:email>someaddress@example.net</eppEAI:email>
C:      </eppEAI:eppEAI>
C:    </extension>
C:    </update>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

5.3. EPP <info> command

When the requested contact object was created using the arbitrary value for the email attribute, the server should return that value for the email attribute.

When the requested contact object was created using the [EAI-DUMMY] value for the email attribute, the server should return the [EAI-DUMMY] value for the email attribute and the corresponding email value in the <eppEAI:email< attribute.

Example <info> command for the authorized client:

```
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:      <contact:infData
S:       xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
S:        <contact:id>sh8013</contact:id>
```

```
S:        <contact:roid>SH8013-REP</contact:roid>
S:        <contact:status s="linked"/>
S:        <contact:status s="clientDeleteProhibited"/>
S:        <contact:postalInfo type="int">
S:          <contact:name>John Doe</contact:name>
S:          <contact:org>Example Inc.</contact:org>
S:          <contact:addr>
S:            <contact:street>123 Example Dr.</contact:street>
S:            <contact:street>Suite 100</contact:street>
S:            <contact:city>Dulles</contact:city>
S:            <contact:sp>VA</contact:sp>
S:            <contact:pc>20166-6503</contact:pc>
S:            <contact:cc>US</contact:cc>
S:          </contact:addr>
S:        </contact:postalInfo>
S:        <contact:voice x="1234">+1.7035555555</contact:voice>
S:        <contact:fax>+1.7035555556</contact:fax>
S:        <contact:email>jdoe@example.com</contact:email>
S:        <contact:clID>ClientY</contact:clID>
S:        <contact:crID>ClientX</contact:crID>
S:        <contact:crDate>1999-04-03T22:00:00.0Z</contact:crDate>
S:        <contact:upID>ClientX</contact:upID>
S:        <contact:upDate>1999-12-03T09:00:00.0Z</contact:upDate>
S:        <contact:trDate>2000-04-08T09:00:00.0Z</contact:trDate>
S:        <contact:authInfo>
S:          <contact:pw>2fooBAR</contact:pw>
S:        </contact:authInfo>
S:        <contact:disclose flag="0">
S:          <contact:voice/>
S:          <contact:email/>
S:        </contact:disclose>
S:      </contact:infData>
S:    </resData>
S:    <extension>
S:      <eppEAI:eppEAI
S:        xmlns:eppEAI=
S:          "urn:ietf:params:xml:ns:epp:eppEAI-1.0">
S:        <eppEAI:email>someaddress@example.com</eppEAI:email>
S:      </eppEAI:eppEAI>
S:    </extension>
```



```
S:    <trID>
```

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

6. Formal syntax

The Internationalized Email Addresses in EPP protocol schema is presented here.

The formal syntax shown here is a complete XML Schema representation of the object mapping suitable for automated validation of EPP XML instances. The <CODE BEGINS> and <CODE ENDS> tags are not part of the XML Schema; they are used to note the beginning and ending of the XML Schema for URI registration purposes.

```
<CODE BEGINS>
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:epp="urn:ietf:params:xml:ns:epp-1.0"
  xmlns:eppcom="urn:ietf:params:xml:ns:eppcom-1.0"
  xmlns:eppEAI="urn:ietf:params:xml:ns:epp:eppEAI-1.0"
  targetNamespace="urn:ietf:params:xml:ns:epp:eppEAI-1.0"
  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>Use of Internationalized Email Addresses in
Extensible Provisioning Protocol v1.0 Schema.</documentation>
  </annotation>
  <!-- Child elements found in EPP commands. -->
  <element name="create" type="eppEAI:eppEAIType" />
  <element name="update" type="eppEAI:eppEAIType" />
  <!--
    The eppEAIType definition
   -->
  <complexType name="eppEAIType">
    <sequence>
      <element name="email"
        type="eppEAI:email" minOccurs="1" maxOccurs = "1" />
    </sequence>
  </complexType>
  <simpleType name="eppEAI:email">
    <restriction base="token" />
  </simpleType>

  <!--
  Child response elements.
  -->
  <element name="infData" type="eppEAI:eppEAIType"/>
</schema>
<CODE ENDS>
```

7. Security Considerations

Registries SHOULD validate the validity of the domain names in the provided email addresses. This can be done by validating all code points according to IDNA2008 [RFC5892].

8. IANA Considerations

8.1. XML Namespace

This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism described in RFC 3688 [RFC3688]. The following URI assignment should be made by IANA:

Registration request for the eppEAI namespace:

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

Registration request for the eppEAI XML Schema:

URI: urn:ietf:params:xml:schema:epp:eppEAI-1.0
Registrant Contact: IESG
XML: See the "Formal Syntax" section of this document.

8.2. EPP Extension Registry

The EPP extension described in this document should be registered by IANA in the "Extensions for the Extensible Provisioning Protocol (EPP)" registry described in RFC 7451 [RFC7451]. The details of the registration are as follows:

Name of Extension: Use of Internationalized Email Addresses
in EPP protocol
Document status: Standards Track
Reference: TBA
Registrant Name and Email Address: IESG, <iesg@ietf.org>
Top-Level Domains(TLDs): Any
IPR Disclosure: None
Status: Active
Notes: None

9. Implementation Considerations

For the sake of uniform syntax on the client side, it is RECOMMENDED to registries to allow any valid address, including the ASCII-only, in the <eppEAI:email> element.

Registries MAY apply extra limitation to the email address syntax (e.g. the addresses can be limited to Left-to-Right scripts). These limitations are out of scope of this document.

10. References

10.1. 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>.

[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>.

[RFC5733] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Contact Mapping", STD 69, RFC 5733, DOI 10.17487/RFC5733, August 2009, <https://www.rfc-editor.org/info/rfc5733>.

[RFC6530] Klensin, J. and Y. Ko, "Overview and Framework for Internationalized Email", RFC 6530, DOI 10.17487/RFC6530, February 2012, <https://www.rfc-editor.org/info/rfc6530>.

[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>.

[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>.

10.2. Informative References

[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>.

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

[RFC5734] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Transport over TCP", STD 69, RFC 5734, DOI 10.17487/RFC5734, August 2009, <https://www.rfc-editor.org/info/rfc5734>.

[RFC5892] Faltstrom, P., Ed., "The Unicode Code Points and Internationalized Domain Names for Applications (IDNA)", RFC 5892, DOI 10.17487/RFC5892, August 2010, <https://www.rfc-editor.org/info/rfc5892>.

10.3. URIs

[1] https://github.com/beldmit/eppeai

Author's Address

Dmitry Belyavskiy
8 marta st.
Moscow
Russian Federation

Phone: +7 916 262 5593
Email: beldmit@gmail.com