Authors: D. Belyavskiy   J. Gould   S. Hollenbeck
                                     VeriSign, Inc.   *Verisign Labs*

RFC 9873
Additional Email Address Extension for the Extensible Provisioning Protocol (EPP)

Abstract

The Extensible Provisioning Protocol (EPP) does not inherently support internationalized email addresses because the specifications for these addresses did not exist when EPP was developed. This document describes a command-response extension that adds support for associating an additional email address with an EPP contact object. That additional email address can be either an internationalized email address or an ASCII-only address.

Status of This Memo

This is an Internet Standards Track document.

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

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9873.

Copyright Notice

Table of Contents

1. Introduction

The framework for internationalized email addresses is described in [RFC6530]. This document describes an Extensible Provisioning Protocol (EPP) [RFC5730] command-response extension that adds support for adding a second email address to the EPP contact object mapping [RFC5733]. The syntax for the email address associated with the base contact object is described in Section 2.6 of [RFC5733]. The second email address can be either an ASCII-only email address or an internationalized SMTPUTF8 email address [RFC6530]. This second address can be used to identify an alternate ASCII-only email address for use in case of primary address delivery issues. It can also be used to identify an SMTPUTF8 address for contact purposes, in which case the ASCII-only address can be used in case of SMTPUTF8 address delivery issues.

While this extension adds support for an additional email address to contact objects, and that additional email address can be an SMTPUTF8 address, it does not in any way update or change any other EPP extension that includes an email address. Adding support for SMTPUTF8 addresses to those extensions will require an update to the relevant extension specifications. In cases where a contact object contains two email addresses, all users of these addresses should be aware that either address may be forwarded to the other. This implies that a message sent to an ASCII-only address may receive a reply from an SMTPUTF8 address or vice versa.

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

The XML namespace prefix "addlEmail" is used for the namespace "urn:ietf:params:xml:ns:epp:addlEmail-1.0", but implementations **MUST NOT** depend on it and instead **MUST** employ a proper namespace-aware XML parser and serializer to interpret and output the XML documents.

2. Email Address Specification

The EPP contact object mapping [RFC5733] normatively references [RFC5322] as the specification for email address syntax. That specification does not include support for internationalized email addresses. [RFC6530] provides an overview and describes the framework for internationalized email. SMTPUTF8 email address syntax is described in Section 3.3 of [RFC6531]. [RFC6531] extends the Mailbox, Local-part, and Domain ABNF rules in [RFC5321] to support "UTF8-non-ascii" (defined in Section 3.1 of [RFC6532]) for the local-part and to support U-label (defined in Section 2.3.2.1 of [RFC5890]) for the domain. The validation rules described in [RFC6531] **MUST** be followed when processing internationalized email addresses associated with this extension.

3. Additional Email Address Element

A second email address can be set using the <addlEmail:addlEmail> element with the command-response extensions defined in Section 5. The <addlEmail:addlEmail> element contains the following child element:

<addlEmail:email>:   An element following the syntax in Section 2 for defining a second ASCII or SMTPUTF8 address. An empty <addlEmail:email/> element unsets the second email address in the Update Command (Section 5.2.5) and indicates the second email is not set in the Info Response (Section 5.1.2). The <addlEmail:email> element contains an **OPTIONAL** "primary" attribute that can be used to indicate that the extension email address should be treated as the primary email address for the extended contact object. The "primary" attribute **MUST NOT** be present if the <addlEmail:email> is empty.

Additional email address considerations:

- The value set for the <contact:disclose><contact:email/> "flag" attribute (described in Section 2.9 of [RFC5733]) **MUST** also be applied to all additional email addresses that are added by a contact extension.
- Any address included in an extension is intended to be an additional address that is associated only with the primary <contact:email> address, and support for any other additional email addresses **MUST** explicitly describe how the additional addresses are associated with the existing addresses.

4. Extension Considerations

4.1. Signaling Client and Server Support

As described in Section 2.4 of [RFC5730], the client and the server can signal support for the extension using a namespace URI in the login and greeting extension services, respectively. The namespace URI "urn:ietf:params:xml:ns:epp:addlEmail-1.0" is used to signal support for the extension. The client includes the namespace URI in an <svcExtension> <extURI> element of the <login> command [RFC5730]. The server includes the namespace URI in an <svcExtension> <extURI> element of the greeting [RFC5730].

4.2. Extension Behavior

4.2.1. Extension Negotiated

If both client and server have indicated support for SMTPUTF8 addresses during session establishment, they **MUST** be able to process an SMTPUTF8 address in any extended contact object during the established EPP session. Server and client obligations when this extension has been successfully negotiated in the EPP session are described below.

The server **MUST** satisfy the following obligations when support for this extension has been negotiated:

- Accept SMTPUTF8-compliant addresses for the extended contact object in the EPP session.
- Support email address validation based on the SMTPUTF8 validation rules defined in Section 2.
- Store email properties that support internationalized characters.
- Return SMTPUTF8-compliant addresses for the extended contact object in EPP responses.
- Support the SMTP extension for internationalized email described in [RFC6531] when sending or receiving email.

The client **MUST** satisfy the following obligations when support for this extension has been negotiated:

- Provide SMTPUTF8-compliant addresses for the extended contact object in the EPP session.
- Accept SMTPUTF8-compliant addresses for the extended contact object in EPP responses.
- Support the SMTP extension for internationalized email described in [RFC6531] when sending or receiving email.

4.2.2. Extension Not Negotiated

An extended contact object **MUST NOT** be provided or returned by either an EPP client or an EPP server when support for this extension is not successfully negotiated at the start of an EPP session.

5. EPP Command Mapping

A detailed description of the EPP syntax and semantics can be found in the EPP core protocol specification [RFC5730]. This section defines the provisioning of an alternate email address.

5.1. EPP Query Commands

EPP provides three commands to retrieve object information: <check> to determine if an object can be provisioned, <info> to retrieve information associated with an object, and <transfer> to retrieve object-transfer status information.

5.1.1. EPP <check> Command

This extension does not add any elements to the EPP <check> command or <check> response described in [RFC5730].

5.1.2. EPP <info> Command

This extension does not add any elements to the EPP <info> command response described in [RFC5730].

If the query is successful, the server replies with an <addlEmail:addlEmail> element (Section 3) along with the regular EPP <resData>.

```
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:      <addlEmail:addlEmail
S:       xmlns:addlEmail="urn:ietf:params:xml:ns:epp:addlEmail-1.0">
S:        <addlEmail:email/>
S:      </addlEmail:addlEmail>
S:    </extension>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:    </trID>
```

```
S:   </response>
S:</epp>
```

Figure 1: Example <info> Contact Response Using the <addlEmail:addlEmail> Extension with No Alternate Email Address

```
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:      <addlEmail:addlEmail
S:       xmlns:addlEmail="urn:ietf:params:xml:ns:epp:addlEmail-1.0">
S:        <addlEmail:email>jdoe-alt@example.net</addlEmail:email>
S:      </addlEmail:addlEmail>
S:    </extension>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:    </trID>
```

```
S:   </response>
S:</epp>
```

Figure 2: Example <info> Contact Response Using the <addlEmail:addlEmail> Extension with an Alternate ASCII Email Address

```
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:      <addlEmail:addlEmail
S:       xmlns:addlEmail="urn:ietf:params:xml:ns:epp:addlEmail-1.0">
S:        <addlEmail:email
S:            primary="true">麥克風@example.com</addlEmail:email>
S:      </addlEmail:addlEmail>
S:    </extension>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:    </trID>
```

```
S:   </response>
S:</epp>
```

Figure 3: Example <info> Contact Response Using the <addlEmail:addlEmail> Extension with an SMTPUTF8 Primary Email Address

5.1.3. EPP <transfer> Query Command

This extension does not add any elements to the EPP <transfer> query command or <transfer> query response described in [RFC5730].

5.2. 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.

5.2.1. EPP <create> Command

This extension defines additional elements to extend the EPP <create> command described in [RFC5733].

The EPP <create> command provides a transform operation that allows a client to create an instance of an object. In addition to the EPP command elements described in [RFC5733], the command **MUST** contain a child <addlEmail:addlEmail> element (Section 3) for the client to set an alternate email address.

```
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>jdoe@example.com</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:     </create>
C:     <extension>
C:       <addlEmail:addlEmail
C:        xmlns:addlEmail="urn:ietf:params:xml:ns:epp:addlEmail-1.0">
C:         <addlEmail:email>jdoe-alt@example.net</addlEmail:email>
C:       </addlEmail:addlEmail>
C:     </extension>
C:     <clTRID>ABC-12345</clTRID>
C:   </command>
C:</epp>
```

Figure 4: Example <create> Command to Create a Contact Object with an Alternate ASCII Email Address

```
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>jdoe@example.com</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:    </create>
C:    <extension>
C:      <addlEmail:addlEmail
C:       xmlns:addlEmail="urn:ietf:params:xml:ns:epp:addlEmail-1.0">
C:        <addlEmail:email
              primary="true">麥克風@example.com</addlEmail:email>
C:      </addlEmail:addlEmail>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

Figure 5: Example <create> Command to Create a Contact Object with a Primary SMTPUTF8 Email Address

This extension does not add any elements to the EPP <create> response described in [RFC5730].

5.2.2. EPP <delete> Command

This extension does not add any elements to the EPP <delete> command or <delete> response described in [RFC5730].

5.2.3. EPP <renew> Command

This extension does not add any elements to the EPP <renew> command or <renew> response described in [RFC5730].

5.2.4. EPP <transfer> Command

This extension does not add any elements to the EPP <transfer> command or <transfer> response described in [RFC5730].

5.2.5. EPP <update> Command

This extension defines additional elements to extend the EPP <update> command described in [RFC5733].

The EPP <update> command provides a transform operation that allows a client to update an instance of an object. In addition to the EPP command elements described in [RFC5733], the command **MUST** contain a child <addlEmail:addlEmail> element (Section 3) for the client to set or unset an alternate email address. If the alternate email address cannot be applied to the object, the server **MUST** return an EPP error result code of 2201.

```
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:update>
C:   </update>
C:   <extension>
C:     <addlEmail:addlEmail
C:      xmlns:addlEmail="urn:ietf:params:xml:ns:epp:addlEmail-1.0">
C:        <addlEmail:email>jdoe-alt@example.net</addlEmail:email>
C:     </addlEmail:addlEmail>
C:   </extension>
C:   <clTRID>ABC-12345</clTRID>
C: </command>
C:</epp>
```

Figure 6: Example <update> Command to Set a Contact Object with an Alternate ASCII Email Address

```
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:update>
C:   </update>
C:   <extension>
C:     <addlEmail:addlEmail
C:      xmlns:addlEmail="urn:ietf:params:xml:ns:epp:addlEmail-1.0">
C:       <addlEmail:email>麥克風@example.com</addlEmail:email>
C:     </addlEmail:addlEmail>
C:   </extension>
C:   <clTRID>ABC-12345</clTRID>
C: </command>
C:</epp>
```

Figure 7: Example <update> Command to Set a Contact Object with an Alternate SMTPUTF8 Email Address

```
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:update>
C:   </update>
C:   <extension>
C:     <addlEmail:addlEmail
C:      xmlns:addlEmail="urn:ietf:params:xml:ns:epp:addlEmail-1.0">
C:       <addlEmail:email/>
C:     </addlEmail:addlEmail>
C:   </extension>
C:   <clTRID>ABC-12345</clTRID>
C: </command>
C:</epp>
```

Figure 8: Example <update> Command to Unset a Contact Object Alternate Email Address

This extension does not add any elements to the EPP <update> response described in [RFC5730].

6. Formal Syntax

The EPP Additional Email Address Extension schema is presented here.

The formal syntax shown here is a complete XML Schema [W3C.REC-xmlschema-1-20041028] [W3C.REC-xmlschema-2-20041028] 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.

6.1. EPP Additional Email Address Extension Schema

```
<CODE BEGINS>
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:addlEmail="urn:ietf:params:xml:ns:epp:addlEmail-1.0"
  targetNamespace="urn:ietf:params:xml:ns:epp:addlEmail-1.0"
  elementFormDefault="qualified">
  <annotation>
    <documentation>Extensible Provisioning Protocol v1.0
        additional email address schema.</documentation>
  </annotation>
  <!-- Create, Update, and Info Response extension element -->
  <element name="addlEmail" type="addlEmail:addlEmailType" />
  <!--
    Single email element that can be empty
   -->
  <complexType name="addlEmailType">
    <sequence>
      <element name="email" type="addlEmail:emailType"/>
    </sequence>
  </complexType>
  <complexType name="emailType">
    <simpleContent>
      <extension base="token">
        <attribute name="primary" type="boolean" default="false"/>
      </extension>
    </simpleContent>
  </complexType>
  <!--
 End of schema.
 -->
</schema>
<CODE ENDS>
```

7. IANA Considerations

7.1. XML Namespace

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

Registration for the addlEmail namespace:

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

Registration for the addlEmail XML Schema:

URI: urn:ietf:params:xml:schema:epp:addlEmail-1.0
Registrant Contact: IESG
XML: See Section 6 of this document.

7.2. EPP Extension Registry

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

Name of Extension: Additional Email Address Extension for the Extensible Provisioning
    Protocol (EPP)
Document Status: Standards Track
Reference: RFC 9873
Registrant Name and Email Address: IESG, <iesg@ietf.org>
TLDs: Any
IPR Disclosure: None
Status: Active
Notes: None

8. Security Considerations

As noted in Sections 10.1 and 13 of [RFC6530], unconstrained Unicode in email addresses can introduce a class of security threats that do not exist with ASCII-only email addresses. As EPP exists in ecosystems where email addresses passed in EPP are displayed in the Registration Data Access Protocol (RDAP) and other services, and copy-and-paste of these email addresses is common for businesses transferring domains via EPP, there should be safeguards against these threats. Therefore, use of the SMTPUTF8 email addresses as described in this document **SHOULD** be done with policies that disallow the use of unconstrained Unicode. The domain-part of these SMTPUTF8 email addresses **SHOULD** conform to IDNA2008 [RFC5895]. The local-part of these SMTPUTF8 email addresses **SHOULD** be restricted to Unicode that does not introduce the threats noted in [RFC6530]. One such possible solution would be to disallow characters outside of Unicode Annex 31 [Unicode-UAX31].

As an email address is often a primary end user contact, an invalid email address may put communication with the end user at risk when such contact is necessary. In case of an invalid domain name in the email address, a malicious actor can register a valid domain name with a similar U-label (homograph attack) and assume control over the domain name associated with the contact using social engineering techniques. To reduce the risk of the use of invalid domain

names in email addresses, registries **SHOULD** validate the domain name syntax in provided email addresses and validate whether the domain name consists of the code points listed in the "IDNA Rules and Derived Property Values" registry <https://www.iana.org/assignments/idna-tables>).

Note that the syntax for internationalized email local-parts is very liberal. Domains are normalized during MX lookup, while local-parts are unconstrained. Implementers may wish to test that their database is able to store difficult local-parts such as U+0061 U+0300 U+00E0. For more on normalization and these three code points, see [RFC5198], Section 3.

9. Privacy Considerations

The content of <addlEmail:email> elements can be processed by EPP clients and servers in the same way that <contact:email> elements are processed, including publication in directory services such as RDAP [STD95]. Many data protection regulations recognize email addresses as personal data, so any policies governing the collection, transmission, and processing of contact information by EPP clients and servers should apply equally to <addlEmail:email> elements.

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>.

[RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, DOI 10.17487/RFC5321, October 2008, <https://www.rfc-editor.org/info/rfc5321>.

[RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, DOI 10.17487/RFC5322, October 2008, <https://www.rfc-editor.org/info/rfc5322>.

[RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, DOI 10.17487/RFC5730, August 2009, <https://www.rfc-editor.org/info/rfc5730>.

[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>.

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, DOI 10.17487/RFC5890, August 2010, <https://www.rfc-editor.org/info/rfc5890>.

[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>.

[RFC6531]   Yao, J. and W. Mao, "SMTP Extension for Internationalized Email", RFC 6531, DOI 10.17487/RFC6531, February 2012, <https://www.rfc-editor.org/info/rfc6531>.

[RFC6532]   Yang, A., Steele, S., and N. Freed, "Internationalized Email Headers", RFC 6532, DOI 10.17487/RFC6532, February 2012, <https://www.rfc-editor.org/info/rfc6532>.

[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>.

[W3C.REC-xmlschema-1-20041028]   Beech, D., Ed., Thompson, H., Ed., Maloney, M., Ed., and N. Mendelsohn, Ed., "XML Schema Part 1: Structures Second Edition", W3C Recommendation, 28 October 2004, <https://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>.

[W3C.REC-xmlschema-2-20041028]   Malhotra, A., Ed. and P. V. Biron, Ed., "XML Schema Part 2: Datatypes Second Edition", W3C Recommendation, 28 October 2004, <https://www.w3.org/TR/2004/REC-xmlschema-2-20041028>.

10.2. Informative References

[RFC5198]   Klensin, J. and M. Padlipsky, "Unicode Format for Network Interchange", RFC 5198, DOI 10.17487/RFC5198, March 2008, <https://www.rfc-editor.org/info/rfc5198>.

[RFC5895]   Resnick, P. and P. Hoffman, "Mapping Characters for Internationalized Domain Names in Applications (IDNA) 2008", RFC 5895, DOI 10.17487/RFC5895, September 2010, <https://www.rfc-editor.org/info/rfc5895>.

[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>.

[STD95]   Internet Standard 95, <https://www.rfc-editor.org/info/std95>.
At the time of writing, this STD comprises the following:

Newton, A., Ellacott, B., and N. Kong, "HTTP Usage in the Registration Data Access Protocol (RDAP)", STD 95, RFC 7480, DOI 10.17487/RFC7480, March 2015, <https://www.rfc-editor.org/info/rfc7480>.

Hollenbeck, S. and N. Kong, "Security Services for the Registration Data Access Protocol (RDAP)", STD 95, RFC 7481, DOI 10.17487/RFC7481, March 2015, <https://www.rfc-editor.org/info/rfc7481>.

Hollenbeck, S. and A. Newton, "Registration Data Access Protocol (RDAP) Query Format", STD 95, RFC 9082, DOI 10.17487/RFC9082, June 2021, <https://www.rfc-editor.org/info/rfc9082>.

Hollenbeck, S. and A. Newton, "JSON Responses for the Registration Data Access Protocol (RDAP)", STD 95, RFC 9083, DOI 10.17487/RFC9083, June 2021, <https://www.rfc-editor.org/info/rfc9083>.

Blanchet, M., "Finding the Authoritative Registration Data Access Protocol (RDAP) Service", STD 95, RFC 9224, DOI 10.17487/RFC9224, March 2022, <https://www.rfc-editor.org/info/rfc9224>.

[Unicode-UAX31]   Davis, M., Ed. and R. Leroy, Ed., "Unicode Identifiers and Syntax", Version 16.0.0, Unicode Standard Annex #31, September 2024, <https://www.unicode.org/reports/tr31/tr31-41.html>. Latest version available at <https://www.unicode.org/reports/tr31/>.

Acknowledgments

Authors' Addresses

Dmitry Belyavskiy
Karpatska 241/3
62500 Brno
Czech Republic
Phone: +420 603 261 036
Email: beldmit@gmail.com

James Gould
VeriSign, Inc.
12061 Bluemont Way
Reston, VA 20190
United States of America
Email: jgould@verisign.com
URI: https://www.verisign.com

Scott Hollenbeck
Verisign Labs
12061 Bluemont Way
Reston, VA 20190
United States of America
Email: shollenbeck@verisign.com
URI: https://www.verisignlabs.com/