Use of Internationalized Email Addresses in the Extensible Provisioning Protocol (EPP)

Abstract

This document describes an EPP extension that permits usage of Internationalized Email Addresses in the EPP protocol and specifies the terms when it can be used by EPP clients and servers. The Extensible Provisioning Protocol (EPP), being developed before the standards for Internationalized Email Addresses (EAI), does not support such email addresses.

TO BE REMOVED on turning to RFC: The document is edited in [the dedicated github repo](). 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 18 February 2023.

Copyright Notice

Table of Contents

1. Introduction

[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 that permits usage of Internationalized Email Addresses in the EPP protocol and specifies the terms when it can be used by EPP clients and servers. A new form of EPP extension, referred to as a Functional Extension, is defined and used to apply the rules for the handling of email address elements in all of the [RFC5730] extensions negotiated in the EPP session, which include the object and command-responses extensions. The described mechanism can be applied to any object or command-response extension that uses an email address.

The Extensible Provisioning Protocol (EPP) specified in [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. This document is only referring to an email address as a property of a managed object, such as the <contact:email> element in the EPP contact mapping [RFC5733] or the <org:email> element in the EPP organization mapping [RFC8543], and command-response extensions applied to a managed object.

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.

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 URI with a higher version number than the prior versions.

3. Email Address Specification

Support of non-ASCII email address syntax is defined in RFC 6530 [RFC6530]. This mapping does not prescribe minimum or maximum lengths for character strings used to represent email addresses. The exact syntax of such addresses is described in Section 3.3 of

[RFC6531]. The validation rules introduced in RFC 6531 MUST be followed when processing this extension.

The definition of email address in the EPP RFCs, including Section 2.6 of [RFC5733] and Section 4.1.2, 4.2.1, and 4.2.5 of [RFC8543], references [RFC5322] for the email address syntax. The XML schema definition in Section 4 of [RFC5733] and Section 5 of [RFC8543] defines the "email" element using the type "eppcom:minTokenType", which is defined in Section 4.2 of [RFC5730] as an XML schema "token" type with minimal length of one. The XML schema "token" type will fully support the use of EAI addresses, so the primary application of the EAI extension is to apply the use of [RFC6531] instead of [RFC5322] for the email address syntax. Other EPP extensions may follow the formal syntax definition using the XML schema type "eppcom:minTokenType" and the [RFC5322] format specification, where this extension applies to all EPP extensions with the same or similar definitions.

The email address format is formally defined in Section 3.4.1 of [RFC5322], which only consists of printable US-ASCII characters for both the local-part and the domain ABNF rules. [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 U-label, defined in Section 2.3.2.1 of [RFC5890], for the domain. By applying the syntax rules of [RFC6531], the EPP extensions will change from supporting only ASCII characters to supporting Internationalized characters both in the email address local-part and domain-part.

4. Functional Extension

[RFC5730] defines three types of extensions at the protocol, object, and command-response level, which impact the structure of the EPP messages. A Functional Extension applies a functional capability to an existing set of EPP extensions and properties. The scope of the applicable EPP extensions and applicable extension properties are defined in the Functional Extension along with the requirements for the servers and clients that support it. The Functional Extension needs to cover the expected behavior of the supporting client or server when interacting with an unsupporting client or server. Negotiating support for a Functional Extension is handled using the EPP Greeting and EPP Login services.

5. Internationalized Email Addresses (EAI) Functional Extension

5.1. Scope of Functional Extension

The functional extension applies to all object extensions and command-response extensions negotiated in the EPP session that

include email address properties. Examples include the
<contact:email> element in the EPP contact mapping [RFC5733] or the
<org:email> element in the EPP organization mapping [RFC8543]. All
registry zones (e.g., top-level domains) authorized for the client
in the EPP session apply. There is no concept of a per-client, per-
zone, per-extension, or per-field setting that is used to indicate
support for EAI, but instead it's a global setting that applies to
the EPP session.

5.2. Signaling Client and Server Support

The client and the server can signal support for the functional
extension using a namespace URI in the login and greeting extension
services respectively. The namespace URI
"urn:ietf:params:xml:ns:epp:eai-1.0" is used to signal support for
the functional extension. 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.

5.3. Functional Extension Behavior

5.3.1. EAI Functional Extension Negotiated

If both client and server have indicated the support of the EAI
addresses during the session establishment, they MUST be able to
process the EAI address in any message having an email property
during the established EPP session. Below are the server and client
obligations when the EAI extension has been successfuly negotiated
in the EPP session.

The server MUST satisfy the following obligations when the EAI
extension has been negotiated:

  *Accept EAI compatible addresses for all email properties in the
   EPP session negotiated object extensions and command-response
   extensions. For example the <contact:email> element in [RFC5733]
   and the <org:email> element in [RFC8543].

  *Accept EAI compatible addresses for all registry zones (e.g.,
   top-level domains) authorized for the client in the EPP session.

  *Email address validation based on EAI validation rules defined in
   Section 3

  *Storage of email properties that support internationalized
   characters.

  *Return EAI compatible addresses for all email properties in the
   EPP responses.

The client MUST satisfy the following obligations when THE EAI
extension has been negotiated:

  *Provide EAI compatible addresses for all e-mail properties in the
   EPP session negotiated object extensions and command-response
   extensions. For example the <contact:email> element in [RFC5733]
   and the <org:email> element in [RFC8543].

  *Provide EAI compatible addresses for all registry zones (e.g.,
   top-level domains) authorized for the client in the EPP session.

  *Accept EAI compatible addresses in the EPP responses for all
   email properties in the EPP session negotiated object extensions
   and command-response extensions.

5.3.2. EAI Functional Extension Not Negotiated

The lack of EAI support can cause data and functional issues, so an
EAI supporting client or server needs to handle cases where the
opposite party doesn't support EAI. Below are the server and client
obligations when the EAI extension is not negotiated due to the lack
of support by the peer.

The EAI supporting server MUST satisfy the following obligations
when the client does not support the EAI extension:

  *When the email property is required in the EPP command, the
   server MUST validate the email property sent by the client using
   the ASCII email validation rules.

  *When the email property is optional in the EPP command, if the
   client supplies the email property the server MUST validate the
   email property using the ASCII email validation rules.

  *When the email property is required in the EPP response, the
   server MUST validate whether the email property is an EAI address
   and if so return the error code 2308 "Data management policy
   violation".

  *When the email property is optional in the EPP response and is
   provided, the server MUST validate whether the email property is
   an EAI address and if so return the error code 2308 "Data
   management policy violation".

The EAI supporting client MUST satisfy the following obligations
when the server does not support the EAI extension:

  *When the email property is required in the EPP command and the
   email property is an EAI address, the client MUST provide an

ASCII email address. The provided email address should provide a
way to contact the registrant.

*When the email property is optional in the EPP command and the
email property is an EAI address and client does not have an
ASCII address providing a way to contact the registrant, the
client MUST omit the email property. If the email property is
provided, the client MUST provide an ASCII email address.

6. IANA Considerations

6.1. XML Namespace

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

Registration request for the eai namespace:

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

6.2. 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

7. 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".

7.1. Verisign EPP SDK

Organization: Verisign Inc.

Name: Verisign EPP SDK

Description: The Verisign EPP SDK includes both a full client
implementation and a full server stub implementation of draft-ietf-
regext-epp-eai.

Level of maturity: Development

Coverage: All aspects of the protocol are implemented.

Licensing: GNU Lesser General Public License

Contact: jgould@verisign.com

URL: https://www.verisign.com/en_US/channel-resources/domain-
registry-products/epp-sdks

8. Security Considerations

The extended security considerations discussion in [RFC6530] and
[RFC6531] applies here.

As email address is often a primary end user contact, an invalid
email address may put the communication with the end user into risk
in case 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 similar U-label (homograph attack) and get a
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 the provided email addresses and validate

whether the domain name consists of the code points allowed by [IDNA Rules and Derived Property Values](#).

When the EAI functional extension is negotiated by both the client and the server, the client and server obligations defined in Section 5.3.1 MUST be satisfied. If the obligations are not satisfied by either the client or server, the EAI address may be mishandled in processing or storage and be unusable.

9. Acknowledgments

The authors would like to thank Alexander Mayrhofer, Chris Lonvick, Gustavo Lozano, Jody Kolker, John C Klensin, John Levine, Klaus Malorny, Marc Blanchet, Marco Schrieck, Mario Loffredo, Murray S. Kucherawy, Patrick Mevzek, Pete Resnick, Scott Hollenbeck, Takahiro Nemoto, Taras Heichenko, and Thomas Corte for their careful review and valuable comments.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.27487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.27487/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 10.27487/RFC5730, August 2009, <https://www.rfc-editor.org/info/rfc5730>.

[RFC5733] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Contact Mapping", STD 69, RFC 5733, DOI 10.27487/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>.

[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 10.27487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

10.2. Informative References

[RFC7451]   Hollenbeck, S., "Extension Registry for the Extensible Provisioning Protocol", RFC 7451, DOI 10.27487/RFC7451, February 2015, <https://www.rfc-editor.org/info/rfc7451>.

[RFC8543]   Zhou, L., Kong, N., Yao, J., Gould, J., and G. Zhou, "Extensible Provisioning Protocol (EPP) Organization Mapping", RFC 8543, DOI 10.27487/RFC8543, March 2019, <https://www.rfc-editor.org/info/rfc8543>.

Appendix A. Change History

A.1. Change from 00 to 01

1. Changed from update of RFC 5733 to use the "Placeholder Text and a New Email Element" EPP Extension approach.

A.2. Change from 01 to 02

1. Fixed the XML schema and the XML examples based on validating them.

2. Added James Gould as co-author.

3. Updated the language to apply to any EPP object mapping and to use the EPP contact mapping as an example.

4. Updated the structure of document to be consistent with the other Command-Response Extensions.

5. Replaced the use of "eppEAI" in the XML namespace and the XML namespace prefix with "eai".

6. Changed to use a pointed XML namespace with "0.2" instead of "1.0".

A.3. Change from 02 to 03

1. The approach has changed to use the concept of Functional EPP Extension.

2. The examples are removed

A.4. Change from 03 to 04

1. More detailed reference to email syntax is provided

2. The shortened eai namespace reference is removed

A.5. Change from 04 to the regext 01 version

1. Provided the recommended placeholder value

A.6. Change from the regext 01 to regext 02 version

1. Removed the concept of the placeholder value

A.7. Change from the regext 02 to regext 03 version

1. Changed to use a pointed XML namespace with "0.3" instead of "0.2".

2. Some wording improvements

A.8. Change from the regext 03 to regext 04 version

1. Some nitpicking

A.9. Change from the regext 04 to regext 05 version

1. Some nitpicking

2. The "Implementation considerations" section is removed

A.10. Change from the regext 05 to regext 06 version

  1. Some nitpicking

A.11. Change from the regext 06 to regext 07 version

  1. Namespace version set to 1.0

A.12. Change from the regext 07 to regext 08 version

  1. Information about implementations is provided.

  2. Acknowledgments section is added.

  3. Reference to RFC 7451 is moved to Informative.

  4. IPR information is provided

  5. Sections are reordered to align with the other regext documents

A.13. Change from the regext 08 to regext 09 version

  1. Nitpicking according to Murray S. Kucherawy review

A.14. Change from the regext 09 to regext 10 version

  1. Some nitpicking in the security considerations.

A.15. Change from the regext 10 to regext 11 version

  1. Nitpicking according mostly GenArt review.

A.16. Change from the regext 11 to regext 12 version

  1. XML schema registration request removed.

A.17. Change from the regext 12 to regext 13 version

  1. Document updated according to SecDir and ART-ART review.

A.18. Change from the regext 13 to regext 14 version

  1. Document updated according the IANA review #1231866.

A.19. Change from the regext 14 to regext 15 version

  1. Document updated according to ART-ART review.

Authors' Addresses

  Dmitry Belyavskiy

8 marta st.
Moscow
127083
Russian Federation

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

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

Email: jgould@verisign.com
URI: http://www.verisigninc.com