Network Working Group
Internet-Draft
Obsoletes: 6068 (if approved)
Intended status: Standards Track
Expires: September 8, 2011

M. Duerst
Aoyama Gakuin University
L. Masinter
Adobe Systems Incorporated
J. Zawinski
DNA Lounge

March 7, 2011

The 'mailto' URI/IRI Scheme
draft-duerst-eai-mailto-00

Abstract

This document defines the format of Uniform Resource Identifiers (URIs) and Internationalized Resource Identfiers (IRIs) to identify resources that are reached using Internet mail. It adds the possibility to use Email Address Internationalization (EAI) email addresses (RFC4952bis) to the previous syntax of 'mailto' URIs (RFC 6068).

Table of Contents

1. Introduction

The 'mailto' URI/IRI scheme [RFC4395bis] is used to identify resources that are reached using Internet mail. In its simplest form, a 'mailto' URI/IRI contains an Internet mail address. For interactions that require message headers or message bodies to be specified, the 'mailto' URI/IRI scheme also allows providing mail header fields and the message body.

This specification extends the previous scheme definition ([RFC6068]) to also allow non-ASCII characters in the left-hand sides (LHSs) of email addresses. To work seamlessly with IRIs ([RFC3987]) and EAI ([RFC4952bis]), these LHSs are percent-encoded based on UTF-8 [STD63] when used in URIs.

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].

In this document, URIs are enclosed in '<' and '>' as described in Appendix C of [STD66]. Extra whitespace and line breaks are added to present long URIs -- they are not part of the actual URI.

2. Syntax of a 'mailto' URI

The syntax of a 'mailto' URI is described using the ABNF of [STD68]. The syntax of a 'mailto' IRI can be obtained from this definition by allowing <iunreserved> characters wherever <unreserved> characters are allowed. The syntax below also uses non-terminal definitions from [STD66] (unreserved, pct-encoded):

```
mailtoURI         = "mailto:" [ to ] [ hfields ]
to                = addr-spec-enc *("," addr-spec-enc )
hfields           = "?" hfield *( "&" hfield )
hfield            = hfname "=" hfvalue
hfname            = *qchar
hfvalue           = *qchar
addr-spec-enc     = local-part-enc "@" domain-enc
local-part-enc    = dot-atom-text-enc / quoted-string-enc
domain-enc        = dot-atom-text-enc / "[" *dtext-no-obs "]"
dtext-no-obs      = %d33-90  ; Printable US-ASCII
                  / %d94-126 ; characters not including
                       ; "[", "]", or "\"
dot-atom-text-enc = <percent-encoded version of
                     dot-atom-text or its EAI equivalent>
quoted-string-enc = <percent-encoded version of
                     dot-atom-text or its EAI equivalent>
qchar             = unreserved / pct-encoded / some-delims
some-delims       = "!" / "$" / "'" / "(" / ")" / "*"
                  / "+" / "," / ";" / ":" / "@"
```

<addr-spec-enc> is a mail address as specified by <addr-spec> in [RFC5322] or <uAddr-Spec> in [RFC5335bis], but excluding <comment>, with the following changes:

1. A number of characters that can appear in <addr-spec> MUST be percent-encoded. These are the characters that cannot appear in a URI according to [STD66] as well as "%" (because it is used for percent-encoding) and all the characters in gen-delims except "@" and ":" (i.e., "/", "?", "#", "[", and "]"). Of the characters in sub-delims, at least the following also have to be percent-encoded: "&", ";", and "=". Care has to be taken both when encoding as well as when decoding to make sure these operations are applied only once.

2. <obs-local-part> and <NO-WS-CTL> as defined in [RFC5322] MUST NOT be used.

3. Whitespace and comments within <local-part-enc> and <domain-enc> MUST NOT be used. They would not have any operational semantics.

4. Percent-encoding can be used in the <domain-enc> part of an <addr-spec-enc>, in order to denote an internationalized domain name. The considerations for <reg-name> in [STD66] apply. In particular, non-ASCII characters MUST first be encoded according to UTF-8 [STD63], and then each octet of the corresponding UTF-8 sequence MUST be percent-encoded to be represented as URI characters. URI-producing applications MUST NOT use percent-encoding in domain names unless it is used to represent a UTF-8

character sequence. When the internationalized domain name is used to compose a message, the name MUST be transformed to the Internationalizing Domain Names in Applications (IDNA) encoding [RFC5891] where appropriate. URI producers SHOULD provide these domain names in the IDNA encoding, rather than percent-encoded, if they wish to maximize interoperability with legacy 'mailto' URI interpreters.

5. Percent-encoding of non-ASCII octets in the <local-part-enc> of an <addr-spec-enc> is used for the internationalization of the <local-part-enc> according to Email Address Internationalization (EAI; [RFC5335bis]). Non-ASCII characters MUST first be encoded according to UTF-8 [STD63], and then each octet of the corresponding UTF-8 sequence MUST be percent-encoded to be represented as URI characters. Any other percent-encoding of non-ASCII characters is prohibited. When a <local-part-enc> containing non-ASCII characters will be used to compose a message, the <local-part-enc> MUST be transformed back to UTF-8 in order to conform to EAI.

<dot-atom-text-enc> is the percent-encoded version of <dot-atom-text> in [RFC5322] or <uDot-Atom-text> in [RFC5335bis]. <quoted-string-enc> is the percent-encoded version of <quoted-string> in [RFC5322] or <uQuoted-String> in [RFC5335bis].

<hfname> and <hfvalue> are encodings of an [RFC5322] header field name and value, respectively. Percent-encoding is needed for the same characters as listed above for <addr-spec-enc>. <hfname> is case-insensitive, but <hfvalue> in general is case-sensitive. Note that [RFC5322] allows all US-ASCII printable characters except ":" in optional header field names (Section 3.6.8), which is the reason why <pct-encoded> is part of the header field name production.

The special <hfname> "body" indicates that the associated <hfvalue> is the body of the message. The "body" field value is intended to contain the content for the first text/plain body part of the message. The "body" pseudo header field is primarily intended for the generation of short text messages for automatic processing (such as "subscribe" messages for mailing lists), not for general MIME bodies. Except for the encoding of characters based on UTF-8 and percent-encoding, no additional encoding (such as e.g., base64 or quoted-printable; see [RFC2045]) is used for the "body" field value. As a consequence, header fields related to message encoding (e.g., Content-Transfer-Encoding) in a 'mailto' URI are irrelevant and MUST be ignored. The "body" pseudo header field name has been registered with IANA for this special purpose (see Section 8.2).

Within 'mailto' URIs, the characters "?", "=", and "&" are reserved,

serving as delimiters. They have to be escaped (as "%3F", "%3D", and "%26", respectively) when not serving as delimiters.

Additional restrictions on what characters are allowed might apply depending on the context where the URI is used. Such restrictions can be addressed by context-specific escaping mechanisms. For example, because the "&" (ampersand) character is reserved in HTML and XML, any 'mailto' URI that contains an ampersand has to be written with an HTML/XML entity ("&") or numeric character reference ("&" or "&").

Non-ASCII characters can be encoded in <hfvalue> as follows:

1. MIME encoded words (as defined in [RFC2047]) are permitted in header field values, but not in an <hfvalue> of a "body" <hfname>. Sequences of characters that look like MIME encoded words can appear in an <hfvalue> of a "body" <hfname>, but in that case have no special meaning. Please note that the '=' and '?' characters used as delimiters in MIME encoded words have to be percent-encoded. Also note that the use of MIME encoded words differs slightly for so-called structured and unstructured header fields.

2. Non-ASCII characters MUST be encoded according to UTF-8 [STD63], and then each octet of the corresponding UTF-8 sequence is percent-encoded to be represented as URI characters. When header field values encoded in this way are used to compose a message conforming to [RFC5322], the <hfvalue> has to be suitably encoded (transformed into MIME encoded words [RFC2047]), except for an <hfvalue> of a "body" <hfname>, which has to be encoded according to [RFC2045]. Please note that for MIME encoded words and for bodies in composed email messages, encodings other than UTF-8 MAY be used as long as the characters are properly transcoded. When header field values encoded in this way are used to compose a message conforming to [RFC5335bis], percent-encoding (including reserved characters) has to be decoded. The header field values can then be used directly because EAI allows UTF-8 in header field values.

Also note that it is syntactically valid to specify both <to> and an <hfname> whose value is "to". That is,

<mailto:addr1@an.example,addr2@an.example>

is equivalent to

<mailto:?to=addr1@an.example,addr2@an.example>

is equivalent to

<mailto:addr1@an.example?to=addr2@an.example>

However, the latter form is NOT RECOMMENDED because different user agents handle this case differently. In particular, some existing clients ignore "to" <hfvalue>s.

Implementations MUST NOT produce two "To:" header fields in a message; the "To:" header field may occur at most once in a message ([RFC5322], Section 3.6). Also, creators of 'mailto' URIs MUST NOT include other message header fields multiple times if these header fields can only be used once in a message.

To avoid interoperability problems, creators of 'mailto' URIs SHOULD NOT use the same <hfname> multiple times in the same URI. If the same <hfname> appears multiple times in a URI, behavior varies widely for different user agents, and for each <hfname>. Examples include using only the first or last <hfname>/<hfvalue> pair, creating multiple header fields, and combining each <hfvalue> by simple concatenation or in a way appropriate for the corresponding header field.

Note that this specification, like any URI scheme specification, does not define syntax or meaning of a fragment identifier (see [STD66]), because these depend on the type of a retrieved representation. In the currently known usage scenarios, a 'mailto' URI cannot be used to retrieve such representations. Therefore, fragment identifiers are meaningless, SHOULD NOT be used on 'mailto' URIs, and SHOULD be ignored upon resolution. The character "#" in <hfvalue>s MUST be escaped as %23.

3. Semantics and Operations

A 'mailto' URI/IRI designates an "Internet resource", which is the mailbox specified in the address. When additional header fields are supplied, the resource designated is the same address but with an additional profile for accessing the resource. While there are Internet resources that can only be accessed via electronic mail, the 'mailto' URI is not intended as a way of retrieving such objects automatically.

The operation of how any URI/IRI scheme is resolved is not mandated by the URI specifications. In current practice, resolving URIs/IRIs such as those in the 'http' URI/IRI scheme causes an immediate interaction between client software and a host running an interactive server. The 'mailto' URI/IRI has unusual semantics because resolving

such a URI/IRI does not cause an immediate interaction with a server. Instead, the client creates a message to the designated address with the various header fields set as default. The user can edit the message, send the message unedited, or choose not to send the message.

The <hfname>/<hfvalue> pairs in a 'mailto' URI/IRI, although syntactically equivalent to header fields in a mail message, do not directly correspond to the header fields in a mail message. In particular, the To, Cc, and Bcc <hfvalue>s don't necessarily result in a header field containing the specified value. Mail client software MAY eliminate duplicate addresses. Creators of 'mailto' URIs SHOULD avoid using the same address twice in a 'mailto' URI/IRI.

Originator fields like From and Date, fields related to routing (Apparently-To, Resent-*, etc.), trace fields, and MIME header fields (MIME-Version, Content-*), when present in the URI/IRI, MUST be ignored. The mail client MUST create new fields when necessary, as it would for any new message. Unrecognized header fields and header fields with values inconsistent with those the mail client would normally send SHOULD be treated as especially suspect. For example, there may be header fields that are totally safe but not known to the MUA, so the MUA MAY choose to show them to the user.

4. Unsafe Header Fields

The user agent interpreting a 'mailto' URI/IRI SHOULD NOT create a message if any of the header fields are considered dangerous; it MAY also choose to create a message with only a subset of the header fields given in the URI/IRI. Only a limited set of header fields such as Subject and Keywords, as well as Body, are believed to be both safe and useful in the general case. In cases where the source of a URI/IRI is well known, and/or specific header fields are limited to specific well-known values, other header fields MAY be considered safe, too.

The creator of a 'mailto' URI/IRI cannot expect the resolver of a URI/IRI to understand more than the "subject" header field and "body". Clients that resolve 'mailto' URIs/IRIs into mail messages MUST be able to correctly create [RFC5322]-compliant mail messages using the "subject" header field and "body".

5. Encoding

[STD66] requires that many characters in URIs/IRIs be encoded. This affects the 'mailto' URI/IRI scheme for some common characters that

might appear in addresses, header fields, or message contents. One such character is space (" ", ASCII hex 20). Note the examples below that use "%20" for space in the message body. Also note that line breaks in the body of a message MUST be encoded with "%0D%0A". Implementations MAY add a final line break to the body of a message even if there is no trailing "%0D%0A" in the body <hfield> of the 'mailto' URI/IRI. Line breaks in other <hfield>s SHOULD NOT be used.

When creating 'mailto' URIs/IRIs, any reserved characters that are used in the URIs/IRIs MUST be encoded so that properly written URI/IRI interpreters can read them. Also, client software that reads URIs/IRIs MUST decode strings before creating the mail message so that the mail message appears in a form that the recipient software will understand. These strings SHOULD be decoded before showing the message to the sending user.

Software creating 'mailto' URIs/IRIs likewise has to be careful to encode any reserved characters that are used. HTML forms are one kind of software that creates 'mailto' URIs/IRIs. Current implementations encode a space as '+', but this creates problems because such a '+' standing for a space cannot be distinguished from a real '+' in a 'mailto' URI/IRI. When producing 'mailto' URIs/IRIs, all spaces SHOULD be encoded as %20, and '+' characters MAY be encoded as %2B. Please note that '+' characters are frequently used as part of an email address to indicate a subaddress, as for example in <bill+ietf@example.org>.

The 'mailto' URI/IRI scheme is limited in that it does not provide for substitution of variables. Thus, it is impossible to create a 'mailto' URI/IRI that includes a user's email address in the message body. This limitation also prevents 'mailto' URIs/IRIs that are signed with public keys and other such variable information.

6. Examples

6.1. Basic Examples

A URI for an ordinary individual mailing address:

<mailto:chris@example.com>

A URI for a mail response system that requires the name of the file to be sent back in the subject:

<mailto:infobot@example.com?subject=current-issue>

A mail response system that requires a "send" request in the body:

<mailto:infobot@example.com?body=send%20current-issue>

A similar URI, with two lines with different "send" requests (in this case, "send current-issue" and, on the next line, "send index"):

<mailto:infobot@
example.com?body=send%20current-issue%0D%0Asend%20index>

An interesting use of 'mailto' URIs occurs when browsing archives of messages. A link can be provided that allows replying to a message and conserving threading information. This is done by adding an In-Reply-To header field containing the Message-ID of the message where the link is added, for example:

<mailto:list@example.org?In-Reply-To=%3C3469A91.D10AF4C@
example.com%3E>

A request to subscribe to a mailing list:

<mailto:majordomo@example.com?body=subscribe%20bamboo-l>

A URI that is for a single user and that includes a CC of another user:

<mailto:joe@example.com?cc=bob@example.com&body=hello>

Note the use of the "&" reserved character above. The following example, using "?" twice, is incorrect:

<mailto:joe@example.com?cc=bob@example.com?body=hello> ; WRONG!

According to [RFC5322], the characters "?", "&", and even "%" may occur in <addr-spec>s. The fact that they are reserved characters is not a problem: those characters may appear in 'mailto' URIs -- they just may not appear in unencoded form. The standard URI encoding mechanisms ("%" followed by a two-digit hex number) MUST be used in these cases.

To indicate the address "gorby%kremvax@example.com" one would use:

<mailto:gorby%25kremvax@example.com>

To indicate the address "unlikely?address@example.com", and include another header field, one would use:

<mailto:unlikely%3Faddress@example.com?blat=foop>

As described above, the "&" (ampersand) character is reserved in HTML

and has to be replaced, e.g., with "&". Thus, a URI with an internal ampersand might look like:

```
Click <a
href="mailto:joe@an.example?cc=bob@an.example&amp;body=hello"
>mailto:joe@an.example?cc=bob@an.example&amp;body=hello</a> to send a
greeting message to Joe and Bob.
```

When an email address itself includes an "&" (ampersand) character, that character has to be percent-encoded. For example, the 'mailto' URI to send mail to "Mike&family@example.org" is <mailto:Mike%26family@example.org>.

6.2. Examples of Complicated Email Addresses

Following are a few examples of how to treat email addresses that contain complicated escaping syntax.

Email address: "not@me"@example.org; corresponding 'mailto' URI:

<mailto:%22not%40me%22@example.org>.

Email address: "oh\\no"@example.org; corresponding 'mailto' URI:

<mailto:%22oh%5C%5Cno%22@example.org>.

Email address: "\\\"it's\ ugly\\\""@example.org; corresponding 'mailto' URI:

<mailto:%22%5C%5C%5C%22it's%5C%20ugly%5C%5C%5C%22%22@example.org>.

6.3. Examples Using UTF-8-Based Percent-Encoding usable with RFC 5322

Sending a mail with the subject "coffee" in French, i.e., "cafe" where the final e is an e-acute, using UTF-8 and percent-encoding:

<mailto:user@example.org?subject=caf%C3%A9>

The same subject, this time using an encoded-word (escaping the "=" and "?" characters used in the encoded-word syntax, because they are reserved):

```
<mailto:user@
example.org?subject=%3D%3Futf-8%3FQ%3Fcaf%3DC3%3DA9%3F%3D>
```

The same subject, this time encoded as iso-8859-1:

<mailto:user@

example.org?subject=%3D%3Fiso-8859-1%3FQ%3Fcaf%3DE9%3F%3D>

Going back to straight UTF-8 and adding a body with the same value:

<mailto:user@example.org?subject=caf%C3%A9&body=caf%C3%A9>

This 'mailto' URI may result in a message looking like this:

```
From: sender@example.net
To: user@example.org
Subject: =?utf-8?Q?caf=C3=A9?=
Content-Type: text/plain;charset=utf-8
Content-Transfer-Encoding: quoted-printable

caf=C3=A9
```

The software sending the email is not restricted to UTF-8, but can use other encodings. The following shows the same email using iso-8859-1 two times:

```
From: sender@example.net
To: user@example.org
Subject: =?iso-8859-1?Q?caf=E9?=
Content-Type: text/plain;charset=iso-8859-1
Content-Transfer-Encoding: quoted-printable

caf=E9
```

Different content transfer encodings (i.e., "8bit" or "base64" instead of "quoted-printable") and different encodings in encoded words (i.e., "B" instead of "Q") can also be used.

For more examples of encoding the word coffee in different languages, see [RFC2324].

The following example uses the Japanese word "natto" (Unicode characters U+7D0D U+8C46) as a domain name label, sending a mail to a user at "natto".example.org:

<mailto:user@%E7%B4%8D%E8%B1%86.example.org?subject=Test&body=NATTO>

When constructing the email for use with [RFC5322], the domain name label is converted to punycode. The resulting message may look as follows:

```
From: sender@example.net
To: user@xn--99zt52a.example.org
Subject: Test
Content-Type: text/plain
Content-Transfer-Encoding: 7bit

NATTO
```

6.4. Examples Using UTF-8-Based Percent-Encoding usable only with EAI

All the previous 'mailto' URIs can be used with EAI. When used with EAI, there is no need to use punycode in domain names, and no need to use MIME encoding in headers and bodies. After decoding percent-encoding, UTF-8 can be used directly. This subsection gives a few additional examples of 'mailto' URI, which can only be used with EAI. Because EAI uses UTF-8 directly, this memo cannot show how a actual constructed message may look.

A hypothetical 'mailto' URI for ordering coffee from a French coffee pot:

mailto:caf%C3%A9@pot.example?Subject=Espresso,%20please

A hypothetical 'mailto' URI for sending a potential erratum to the first author of this memo ("%C3%BC" represents an u-umlaut, "%E9%9D%92%E5%B1%B1" represents the Unicode characters U+9752 (blue) and U+5C71 (mountain)):

mailto:Martin.D%C3%BCrst@%E9%9D%92%E5%B1%B1.ac.jp?Subject=Error%20in%20RFC6068bis

7. Security Considerations

The 'mailto' URI/IRI scheme can be used to send a message from one user to another, and thus can introduce many security concerns. Mail messages can be logged at the originating site, the recipient site, and intermediary sites along the delivery path. If the messages are not encrypted, they can also be read at any of those sites.

A 'mailto' URI/IRI gives a template for a message that can be sent by mail client software. The contents of that template may be opaque or difficult to read by the user at the time of specifying the URI/IRI, as well as being hidden in the user interface (for example, a link on

an HTML Web page might display something other than the content of the corresponding 'mailto' URI/IRI that would be used when clicked). Thus, a mail client SHOULD NOT send a message based on a 'mailto' URI/IRI without first disclosing and showing to the user the full message that will be sent (including all header fields that were specified by the 'mailto' URI/IRI), fully decoded, and asking the user for approval to send the message as electronic mail. The mail client SHOULD also make it clear that the user is about to send an electronic mail message, since the user may not be aware that this is the result of a 'mailto' URI/IRI. Users are strongly encouraged to ensure that the 'mailto' URI/IRI presented to them matches the address included in the "To:" line of the email message.

Some header fields are inherently unsafe to include in a message generated from a URI/IRI. For details, please see Section 3. In general, the fewer header fields interpreted from the URI/IRI, the less likely it is that a sending agent will create an unsafe message.

Examples of problems with sending unapproved mail include:

- mail that breaks laws upon delivery, such as making illegal threats;
- mail that identifies the sender as someone interested in breaking laws;
- mail that identifies the sender to an unwanted third party;
- mail that causes a financial charge to be incurred by the sender;
- mail that causes an action on the recipient machine that causes damage that might be attributed to the sender.

Programs that interpret 'mailto' URIs/IRIs SHOULD ensure that the SMTP envelope return path address, which is given as an argument to the SMTP MAIL FROM command, is set and correct, and that the resulting email is a complete, workable message.

'mailto' URIs/IRIs on public Web pages expose mail addresses for harvesting. This applies to all mail addresses that are part of the 'mailto' URI/IRI, including the addresses in a "bcc" <hfvalue>. Those addresses will not be sent to the recipients in the 'to' field and in the "to" and "cc" <hfvalue>s, but will still be publicly visible in the URI/IRI. Addresses in a "bcc" <hfvalue> may also leak to other addresses in the same <hfvalue> or become known otherwise, depending on the mail user agent used.

Programs manipulating 'mailto' URIs/IRIs have to take great care to

not inadvertently double-escape or double-unescape 'mailto' URIs/IRIs, and to make sure that escaping and unescaping conventions relating to URIs/IRIs and relating to mail addresses are applied in the right order.

Implementations parsing 'mailto' URIs/IRIs must take care to sanity check 'mailto' URIs/IRIs in order to avoid buffer overflows and problems resulting from them (e.g., execution of code specified by the attacker).

The security considerations of [STD66], [RFC5890], [RFC5891], and [RFC3987] also apply. Implementers and users are advised to check them carefully.

8. IANA Considerations

8.1. Update of the Registration of the 'mailto' URI/IRI Scheme

This document changes the definition of the 'mailto' URI/IRI scheme; the registry of URI/IRI schemes should be updated to refer to this document (RFC YYYY) rather than its predecessor, [RFC6068]. The registration template is as follows:

URI scheme name:
'mailto'

Status:
permanent

URI/IRI scheme syntax:
See the syntax section of RFC YYYY.

URI/IRI scheme semantics:
See the semantics section of RFC YYYY.

Encoding considerations:
See the syntax and encoding sections of RFC YYYY.

Applications/protocols that use this URI scheme name:
The 'mailto' URI/IRI scheme is widely used since the start of the Web.

Interoperability considerations:
Interoperability for 'mailto' URIs/IRIs with UTF-8-based percent-encoding might be somewhat lower than interoperability for 'mailto' URIs with US-ASCII only. In particular, interoperability for 'mailto' URIs/IRIs with UTF-8-based percent-encoding in the LHS of email addresses requires support of EAI to work.

Security considerations:
See the security considerations section of RFC YYYY.

Contact:
IETF

Author/Change controller:
IETF

References:
RFC YYYY

8.2. Registration of the Body Header Field

IANA is herewith requested to update the reference for the registration of the Body header field in the Message Header Fields Registry ([RFC3864]) from [RFC6068] to this document (there are no changes to the specification of the Body header field itself).

9. Change record

9.1. Main Changes from RFC 6068

The main changes from [RFC6068] are as follows:

- Allowed UTF-8/percent-encoding in <local-part-enc>, to be used for EAI email addresses.
- Added suffix "-enc" to some ABNF rule names to distinguish them from their counterparts without percent-encoding.
- Added a MUST for using UTF-8 in <hfvalue>.

9.2. Changes from RFC 6068 to -00

Changed title and various other places to also refer to IRIs.

Updated syntax to use "-enc" prefix in some places.

Added MUST for using UTF-8 in <hfvalue>.

Added a new subsection with EAI-only examples.

Updated references.

Updated first author's address.

10. Acknowledgments

This document was derived from [RFC6068]; the acknowledgments from that specification and its predecessor still apply.

Valuable input on this document was received from (in no particular order): [names to be added when we get comments].

11. References

11.1. Normative References

[RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

[RFC2047] Moore, K., "MIME Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, November 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3864] Klyne, G., Nottingham, M., and J. Mogul, "Registration Procedures for Message Header Fields", BCP 90, RFC 3864, September 2004.

[RFC3987] Duerst, M. and M. Suignard, "Internationalized Resource Identifiers (IRIs)", RFC 3987, January 2005.

[RFC5322] Resnik, P., "Internet Message Format", RFC 5322, October 2008.

[RFC5335bis] Abel, Y. and S. Steele, "Internationalized Email Headers", draft draft-ietf-eai-rfc5335bis-09, March 2011.

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.

[RFC5891] Klensin, J., "Internationalized Domain Names in Applications (IDNA): Protocol", RFC 5891, August 2010.

[STD63] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

[STD66] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[STD68] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

11.2. Informative References

[RFC2324] Masinter, L., "Hyper Text Coffee Pot Control Protocol (HTCPCP/1.0)", RFC 2324, April 1998.

[RFC4395bis]
Hansen, T., Hardie, T., and L. Masinter, "Guidelines and Registration Procedures for New URI/IRI Schemes", draft draft-ietf-iri-4395bis-irireg-00, October 2010.

[RFC4952bis]
Klensin, J. and Y. Ko, "Overview and Framework for Internationalized Email", draft draft-ietf-eai-frmwrk-4952bis-10, September 2010.

[RFC6068] Duerst, M., Masinter, L., and J. Zawinski, "The mailto URL scheme", RFC 6068, October 2010.

Authors' Addresses

Martin Duerst (Note: Please write "Duerst" with u-umlaut wherever possible, for example as "Dürst" in XML and HTML.)
Aoyama Gakuin University
5-10-1 Fuchinobe
Chuo-ku
Sagamihara, Kanagawa 252-5258
Japan

Phone: +81 42 759 6329
Fax: +81 42 759 6495
Email: duerst@it.aoyama.ac.jp
URI: http://www.sw.it.aoyama.ac.jp/D%C3%BCrst/

Larry Masinter
Adobe Systems Incorporated
345 Park Ave
San Jose, CA 95110
USA

Phone: +1-408-536-3024
Email: LMM@acm.org
URI: http://larry.masinter.net/

Jamie Zawinski
DNA Lounge
375 Eleventh Street
San Francisco, CA 94103
USA

Email: jwz@jwz.org

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: October 1, 2011

P. Resnick, Ed.
Qualcomm Incorporated
C. Newman, Ed.
Oracle
S. Shen, Ed.
CNNIC

March 30, 2011

IMAP Support for UTF-8
draft-ietf-eai-5738bis-00

Abstract

This specification extends the Internet Message Access Protocol version 4rev1 (IMAP4rev1) to support UTF-8 encoded international characters in user names, mail addresses and message headers.

Status of This Memo

This Internet-Draft is submitted to IETF 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 October 1, 2011.

Copyright Notice

described in the Simplified BSD License.

Table of Contents

1. Introduction

This specification extends IMAP4rev1 [RFC3501] to permit UTF-8 [RFC3629] in headers as described in "Internationalized Email Headers" [RFC5335]. It also adds a mechanism to support mailbox names, login names, and passwords using the UTF-8 charset. This specification creates five new IMAP capabilities to allow servers to advertise these new extensions, along with two new IMAP LIST selection options and a new IMAP LIST return option.

This specification permits implementation of an IMAP server that hides mailboxes with internationalized email messages from IMAP clients that do not support this extension. Implementation of "Post-delivery Message Downgrading for Internationalized Email Messages" [popimap-downgrade] is necessary for an IMAP server to make mailboxes with internationalized email messages visible to IMAP clients that do not support this extension.

2. Conventions Used in this Document

The key words "MUST", "MUST NOT", "SHOULD", "SHOULD NOT", and "MAY" in this document are to be interpreted as defined in "Key words for use in RFCs to Indicate Requirement Levels" [RFC2119].

The formal syntax uses the Augmented Backus-Naur Form (ABNF) [RFC5234] notation including the core rules defined in Appendix B of [RFC5234]. In addition, rules from IMAP4rev1 [RFC3501], UTF-8 [RFC3629], "Collected Extensions to IMAP4 ABNF" [RFC4466], and IMAP4 LIST Command Extensions [RFC5258] are also referenced.

In examples, "C:" and "S:" indicate lines sent by the client and server, respectively. If a single "C:" or "S:" label applies to multiple lines, then the line breaks between those lines are for editorial clarity only and are not part of the actual protocol exchange.

3. UTF8=ACCEPT IMAP Capability

The "UTF8=ACCEPT" capability indicates that the server supports UTF-8 quoted strings, the "UTF8" parameter to SELECT and EXAMINE, and UTF-8 responses from the LIST and LSUB commands.

A client MUST use the "ENABLE UTF8=ACCEPT" command (defined in [RFC5161]) to indicate to the server that the client accepts UTF-8 quoted-strings. The "ENABLE UTF8=ACCEPT" command MUST only be used in the authenticated state. (Note that the "UTF8=ONLY" capability described in Section 7 and the "UTF8=ALL" capability described in Section 6 imply the "UTF8=ACCEPT" capability. See additional

information in these sections.)

3.1. IMAP UTF-8 Quoted Strings

The IMAP4rev1 [RFC3501] base specification forbids the use of 8-bit characters in atoms or quoted strings. Thus, a UTF-8 string can only be sent as a literal. This can be inconvenient from a coding standpoint, and unless the server offers IMAP4 non-synchronizing literals [RFC2088], this requires an extra round trip for each UTF-8 string sent by the client. When the IMAP server advertises the "UTF8=ACCEPT" capability, it informs the client that it supports native UTF-8 quoted-strings with the following syntax:

```
string        =/ uQuoted

uQuoted       = "*" DQUOTE *uQUOTED-CHAR DQUOTE

    DQUOTE        = <Defined in appendix B.1 of RFC 5234>

uQUOTED-CHAR  = QUOTED-CHAR / UTF8-2 / UTF8-3 / UTF8-4

    UTF8-2        =   <Defined in Section 4 of RFC3629>

UTF8-3        =   <Defined in Section 4 of RFC3629>

UTF8-4        =   <Defined in Section 4 of RFC3629>
```

When this quoting mechanism is used by the client (specifically an octet sequence beginning with *" and ending with "), then the server MUST reject octet sequences with the high bit set that fail to comply with the formal syntax in [RFC3629] with a BAD response.

The IMAP server MUST NOT send utf8-quoted syntax to the client unless the client has indicated support for that syntax by using the "ENABLE UTF8=ACCEPT" command.

If the server advertises the "UTF8=ACCEPT" capability, the client MAY use utf8-quoted syntax with any IMAP argument that permits a string (including astring and nstring). However, if characters outside the US-ASCII repertoire are used in an inappropriate place, the results would be the same as if other syntactically valid but semantically invalid characters were used. For example, if the client includes UTF-8 characters in the user or password arguments (and the server has not advertised "UTF8=USER"), the LOGIN command will fail as it would with any other invalid user name or password. Specific cases where UTF-8 characters are permitted or not permitted are described in the following paragraphs.

All IMAP servers that advertise the "UTF8=ACCEPT" capability SHOULD accept UTF-8 in mailbox names, and those that also support the "Mailbox International Naming Convention" described in RFC 3501, Section 5.1.3 MUST accept utf8-quoted mailbox names and convert them to the appropriate internal format. Mailbox names MUST comply with the Net-Unicode Definition (Section 2 of [RFC5198]) with the specific exception that they MUST NOT contain control characters (0000-001F, 0080-009F), delete (007F), line separator (2028), or paragraph separator (2029).

An IMAP client MUST NOT issue a SEARCH command that uses a mixture of utf8-quoted syntax and a SEARCH CHARSET other than UTF-8. If an IMAP server receives such a SEARCH command, it SHOULD reject the command with a BAD response (due to the conflicting charset labels).

3.2. UTF8 Parameter to SELECT and EXAMINE

The "UTF8=ACCEPT" capability also indicates that the server supports the "UTF8" parameter to SELECT and EXAMINE. When a mailbox is selected with the "UTF8" parameter, it alters the behavior of all IMAP commands related to message sizes, message headers, and MIME body headers so they refer to the message with UTF-8 headers. If the mailstore is not UTF-8 header native and the SELECT or EXAMINE command with UTF-8 header modifier succeeds, then the server MUST return results as if the mailstore were UTF-8 header native with upconversion requirements as described in Section 8. The server MAY reject the SELECT or EXAMINE command with the [NOT-UTF-8] response code, unless the "UTF8=ALL" or "UTF8=ONLY" capability is advertised.

Servers MAY include mailboxes that can only be selected or examined if the "UTF8" parameter is provided. However, such mailboxes MUST NOT be included in the output of an unextended LIST, LSUB, or equivalent command. If a client attempts to SELECT or EXAMINE such mailboxes without the "UTF8" parameter, the server MUST reject the command with a [UTF-8-ONLY] response code. As a result, such mailboxes will not be accessible by IMAP clients written prior to this specification and are discouraged unless the server advertises "UTF8=ONLY" or the server implements IMAP4 LIST Command Extensions [RFC5258].

```
utf8-select-param = "UTF8" ;; Conforms to select-param from RFC 4466
```

```
C: a SELECT newmailbox (UTF8)
```

```
S: ...
```

S: a OK SELECT completed

C: b FETCH 1 (SIZE ENVELOPE BODY)

S: ... UTF-8 header native results

S: b OK FETCH completed

C: c EXAMINE legacymailbox (UTF8)

S: c NO [NOT-UTF-8] Mailbox does not support UTF-8 access

C: d SELECT funky-new-mailbox

S: d NO [UTF-8-ONLY] Mailbox requires UTF-8 client

3.3. UTF-8 LIST and LSUB Responses

After an IMAP client successfully issues an "ENABLE UTF8=ACCEPT" command, the server MUST NOT return in LIST results any mailbox names to the client following the IMAP4 Mailbox International Naming Convention. Instead, the server MUST return any mailbox names with characters outside the US-ASCII repertoire using utf8-quoted syntax. (The IMAP4 Mailbox International Naming Convention has proved problematic in the past, so the desire is to make this syntax obsolete as quickly as possible.)

3.4. UTF-8 Interaction with IMAP4 LIST Command Extensions

When an IMAP server advertises both the "UTF8=ACCEPT" capability and the "LIST-EXTENDED" [RFC5258] capability, the server MUST support the LIST extensions described in this section.

3.4.1. UTF8 and UTF8ONLY LIST Selection Options

The "UTF8" LIST selection option tells the server to include mailboxes that only support UTF-8 headers in the output of the list command. The "UTF8ONLY" LIST selection option tells the server to include all mailboxes that support UTF-8 headers and to exclude mailboxes that don’t support UTF-8 headers. Note that "UTF8ONLY"

implies "UTF8", so it is not necessary for the client to request both. Use of either selection option will also result in UTF-8 mailbox names in the result as described in Section 3.3 and implies the "UTF8" List return option described in Section 3.4.2.

3.4.2. UTF8 LIST Return Option

If the client supplies the "UTF8" LIST return option, then the server MUST include either the "\NoUTF8" or the "\UTF8Only" mailbox attribute as appropriate. The "\NoUTF8" mailbox attribute indicates that an attempt to SELECT or EXAMINE that mailbox with the "UTF8" parameter will fail with a [NOT-UTF-8] response code. The "\UTF8Only" mailbox attribute indicates that an attempt to SELECT or EXAMINE that mailbox without the "UTF8" parameter will fail with a [UTF-8-ONLY] response code. Note that computing this information may be expensive on some server implementations, so this return option should not be used unless necessary.

The ABNF [RFC5234] for these LIST extensions follows:

```
uList-select-independent-opt =/ "UTF8"

list-select-base-opt        =/ "UTF8ONLY"

mbx-list-oflag              =/ "\NoUTF8" / "\UTF8Only"

return-option               =/ "UTF8"

resp-text-code              =/ "NOT-UTF-8" / "UTF-8-ONLY"
```

4. UTF8=APPEND Capability

If the "UTF8=APPEND" capability is advertised, then the server accepts UTF-8 headers in the APPEND command message argument. A client that sends a message with UTF-8 headers to the server MUST send them using the "UTF8" APPEND data extension. If the server also advertises the CATENATE capability (as specified in [RFC4469]), the client can use the same data extension to include such a message in a CATENATE message part. The ABNF for the APPEND data extension and CATENATE extension follows:

```
utf8-literal   = "UTF8" SP "(" literal8 ")"

append-data    =/ utf8-literal

cat-part       =/ utf8-literal
```

A server that advertises "UTF8=APPEND" MAY fail for \NotUTF8

mailboxes with a NOT-UTF-8 response code. If this command does not fail, it MAY follow the requirements of the IMAP base specification and [RFC5322] for message fetching. Mechanisms for 7-bit downgrading to help comply with the standards are discussed in [popimap-downgrade].

IMAP servers that do not advertise the "UTF8=APPEND" or "UTF8=ONLY" capability SHOULD reject an APPEND command that includes any 8-bit in the message headers with a "NO" response.

Note that the "UTF8=ONLY" capability described in Section 7 implies the "UTF8=APPEND" capability. See additional information in that section.

5. UTF8=USER Capability

If the "UTF8=USER" capability is advertised, that indicates the server accepts UTF-8 user names and passwords and applies SASLprep [RFC4013] to both arguments of the LOGIN command. The server MUST reject UTF-8 that fails to comply with the formal syntax in RFC 3629 [RFC3629] or if it encounters Unicode characters listed in Section 2.3 of SASLprep RFC 4013 [RFC4013].

6. UTF8=ALL Capability

The "UTF8=ALL" capability indicates all server mailboxes support UTF-8 headers. Specifically, SELECT and EXAMINE with the "UTF8" parameter will never fail with a [NOT-UTF-8] response code.

Note that the "UTF8=ONLY" capability described in Section 7 implies the "UTF8=ALL" capability. See additional information in that section.

Note that the "UTF8=ALL" capability implies the "UTF8=ACCEPT" capability.

7. UTF8=ONLY Capability

The "UTF8=ONLY" capability permits an IMAP server to advertise that it does not support the international mailbox name convention (modified UTF-7), and does not permit selection or examination of any mailbox unless the "UTF8" parameter is provided. As this is an incompatible change to IMAP, a clear warning is necessary. IMAP clients that find implementation of the "UTF8=ONLY" capability problematic are encouraged to at least detect the "UTF8=ONLY" capability and provide an informative error message to the end-user.

The "UTF8=ONLY" capability implies the "UTF8=ACCEPT" capability, the

"UTF8=ALL" capability, and the "UTF8=APPEND" capability. A server that advertises "UTF8=ONLY" need not advertise the three implicit capabilities.

8. Up-Conversion Server Requirements

When an IMAP4 server uses a traditional mailbox format that includes 7-bit headers and it chooses to permit access to that mailbox with the "UTF8" parameter, it MUST support minimal up-conversion as described in this section.

The server MUST support up-conversion of the following address header-fields in the message header: From, Sender, To, CC, Bcc, Resent-From, Resent-Sender, Resent-To, Resent-CC, Resent-Bcc, and Reply-To. This up-conversion MUST include address domains encoded according to Internationalizing Domain Names in Applications (IDNA) [RFC5890], and MIME header encoding [RFC2047] of display-names and any [RFC5322] comments.

The following charsets MUST be supported for up-conversion of MIME header encoding [RFC2047]: UTF-8, US-ASCII, ISO-8859-1, ISO-8859-2, ISO-8859-3, ISO-8859-4, ISO-8859-5, ISO-8859-6, ISO-8859-7, ISO-8859-8, ISO-8859-9, ISO-8859-10, ISO-8859-14, and ISO-8859-15. If the server supports other charsets in IMAP SEARCH or IMAP CONVERT [RFC5259], it SHOULD also support those charsets in this conversion.

Up-conversion of MIME header encoding of the following headers MUST also be implemented: Subject, Date ([RFC5322] comments only), Comments, Keywords, and Content-Description.

Server implementations also SHOULD up-convert all MIME body headers [RFC2045], SHOULD up-convert or remove the deprecated (and misused) "name" parameter [RFC1341] on Content-Type, and MUST up-convert the Content-Disposition [RFC2183] "filename" parameter, except when any of these are contained within a multipart/signed MIME body part (see below). These parameters can be encoded using the standard MIME parameter encoding [RFC2231] mechanism, or via non-standard use of MIME header encoding [RFC2047] in quoted strings.

The IMAP server MUST NOT perform up-conversion of headers and content of multipart/signed, as well as Original-Recipient and Return-Path.

9. Issues with UTF-8 Header Mailstore

When an IMAP server uses a mailbox format that supports UTF-8 headers and it permits selection or examination of that mailbox without the "UTF8" parameter, it is the responsibility of the server to comply with the IMAP4rev1 base specification [RFC3501] and [RFC5322] with

respect to all header information transmitted over the wire. Mechanisms for 7-bit downgrading to help comply with the standards are discussed in [popimap-downgrade].

An IMAP server with a mailbox that supports UTF-8 headers MUST comply with the protocol requirements implicit from Section 8. However, the code necessary for such compliance need not be part of the IMAP server itself in this case. For example, the minimal required up-conversion could be performed when a message is inserted into the IMAP-accessible mailbox.

10. IANA Considerations

This adds five new capabilities ("UTF8=ACCEPT", "UTF8=USER", "UTF8=APPEND", "UTF8=ALL", and "UTF8=ONLY") to the IMAP4rev1 Capabilities registry [RFC3501].

This adds two new IMAP4 list selection options and one new IMAP4 list return option.

1. LIST-EXTENDED option name: UTF8

   LIST-EXTENDED option type: SELECTION

   Implied return options(s): UTF8

   LIST-EXTENDED option description: Causes the LIST response to include mailboxes that mandate the UTF8 SELECT/EXAMINE parameter.

   Published specification: RFC 5738bis, Section 3.4.1

   Security considerations: RFC 5738bis, Section 11

   Intended usage: COMMON

   Person and email address to contact for further information: see the Authors' Addresses at the end of this specification

   Owner/Change controller: iesg@ietf.org

2. LIST-EXTENDED option name: UTF8ONLY

   LIST-EXTENDED option type: SELECTION

   Implied return options(s): UTF8

   LIST-EXTENDED option description: Causes the LIST response to include mailboxes that mandate the UTF8 SELECT/EXAMINE parameter

and exclude mailboxes that do not support the UTF8 SELECT/EXAMINE parameter.

Published specification: RFC 5738bis, Section 3.4.1

Security considerations: RFC 5738bis, Section 11

Intended usage: COMMON

Person and email address to contact for further information: see the Authors' Addresses at the end of this specification

Owner/Change controller: iesg@ietf.org

3. LIST-EXTENDED option name: UTF8

LIST-EXTENDED option type: RETURN

Implied return options(s): none

LIST-EXTENDED option description: Causes the LIST response to include \NoUTF8 and \UTF8Only mailbox attributes.

Published specification: RFC 5738bis, Section 3.4.1

Security considerations: RFC 5738bis, Section 11

Intended usage: COMMON

Person and email address to contact for further information: see the Authors' Addresses at the end of this specification

Owner/Change controller: iesg@ietf.org

11. Security Considerations

The security considerations of UTF-8 [RFC3629] and SASLprep [RFC4013] apply to this specification, particularly with respect to use of UTF-8 in user names and passwords. Otherwise, this is not believed to alter the security considerations of IMAP4rev1.

[**]

This document does not address downgrading scenarios, the security issues are discussed in [popimap-downgrade]

12. References

12.1. Normative References

[RFC1341] Borenstein, N. and N. Freed, "MIME (Multipurpose Internet Mail Extensions): Mechanisms for Specifying and Describing the Format of Internet Message Bodies", RFC 1341, June 1992.

[RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

[RFC2047] Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, November 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2183] Troost, R., Dorner, S., and K. Moore, "Communicating Presentation Information in Internet Messages: The Content-Disposition Header Field", RFC 2183, August 1997.

[RFC2231] Freed, N. and K. Moore, "MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations", RFC 2231, November 1997.

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.

[RFC3501] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", RFC 3501, March 2003.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

[RFC4013] Zeilenga, K., "SASLprep: Stringprep Profile for User Names and Passwords", RFC 4013, February 2005.

[RFC4466] Melnikov, A. and C. Daboo, "Collected Extensions to IMAP4 ABNF", RFC 4466, April 2006.

[RFC4469] Resnick, P., "Internet Message Access Protocol (IMAP) CATENATE Extension", RFC 4469, April 2006.

[RFC5161] Gulbrandsen, A. and A. Melnikov, "The IMAP ENABLE Extension", RFC 5161, March 2008.

[RFC5198] Klensin, J. and M. Padlipsky, "Unicode Format for Network Interchange", RFC 5198, March 2008.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC5258] Leiba, B. and A. Melnikov, "Internet Message Access Protocol version 4 - LIST Command Extensions", RFC 5258, June 2008.

[RFC5259] Melnikov, A. and P. Coates, "Internet Message Access Protocol - CONVERT Extension", RFC 5259, July 2008.

[RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, October 2008.

[RFC5335] Abel, Y., "Internationalized Email Headers", RFC 5335, September 2008.

[RFC5738] Resnick, P. and C. Newman, "IMAP Support for UTF-8", RFC 5738, March 2010.

12.2. Informative References

[RFC2049] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples", RFC 2049, November 1996.

[RFC2088] Myers, J., "IMAP4 non-synchronizing literals", RFC 2088, January 1997.

[RFC2277] Alvestrand, H., "IETF Policy on Character Sets and Languages", BCP 18, RFC 2277, January 1998.

[RFC5721] Gellens, R. and C. Newman, "POP3 Support for UTF-8", RFC 5721, February 2010.

[popimap-downgrade] Fujiwara, K., "Post-delivery Message Downgrading

for Internationalized Email Messages", draft-ietf-eai-popimap-downgrade-00 (work in progress), October 2010.

Appendix A. Appendix A. Design Rationale

This non-normative section discusses the reasons behind some of the design choices in the above specification.

The basic approach of advertising the ability to access a mailbox in UTF-8 mode is intended to permit graceful upgrade, including servers that support multiple mailbox formats. In particular, it would be undesirable to force conversion of an entire server mailstore to UTF-8 headers, so being able to phase-in support for new mailboxes and gradually migrate old mailboxes is permitted by this design.

"UTF8=USER" is optional because many identity systems are US-ASCII only, so it’s helpful to inform the client up front that UTF-8 won’t work.

The "UTF8=ONLY" mechanism simplifies diagnosis of interoperability problems when legacy support goes away. In the situation where backwards compatibility is broken anyway, just-send-UTF-8 IMAP has the advantage that it might work with some legacy clients. However, the difficulty of diagnosing interoperability problems caused by a just-send-UTF-8 IMAP mechanism is the reason the "UTF8=ONLY" capability mechanism was chosen.

The up-conversion requirements are designed to balance the desire to deprecate and eventually eliminate complicated encodings (like MIME header encodings) without creating a significant deployment burden for servers. As IMAP4 servers already require a MIME parser, this includes additional server up-conversion requirements not present in POP3 Support for UTF-8 [RFC5721].

The set of mandatory charsets comes from two sources: MIME requirements [RFC2049] and IETF Policy on Character Sets [RFC2277]. Including a requirement to up-convert widely deployed encoded ideographic charsets to UTF-8 would be reasonable for most scenarios, but may require unacceptable table sizes for some embedded devices. The open-ended recommendation to support widely deployed charsets avoids the political ramifications of attempting to list such charsets. The authors believe market forces, existing open-source software, and public conversion tables are sufficient to deploy the appropriate charsets.

Appendix B. Appendix B. Examples Demonstrating Relationships between UTF8= Capabilities

UTF8=ACCEPT UTF8=USER UTF8=APPEND

UTF8=ACCEPT UTF8=ALL

UTF8=ALL ; Note, same as above

UTF8=ACCEPT UTF8=USER UTF8=APPEND UTF8=ALL UTF8=ONLY

UTF8=USER UTF8=ONLY ; Note, same as above

Appendix C. Appendix C. Acknowledgments

The authors wish to thank the participants of the EAI working group for their contributions to this document with particular thanks to Harald Alvestrand, David Black, Randall Gellens, Arnt Gulbrandsen, Kari Hurtta, John Klensin, Xiaodong Lee, Charles Lindsey, Alexey Melnikov, Subramanian Moonesamy, Shawn Steele, Daniel Taharlev, and Joseph Yee for their specific contributions to the discussion.

Authors' Addresses

Pete Resnick (editor)
Qualcomm Incorporated
5775 Morehouse Drive
San Diego, CA 92121-1714
US

Phone: +1 858 651 4478
EMail: presnick@qualcomm.com

Chris Newman (editor)
Oracle
800 Royal Oaks
Monrovia, CA 91016
USA

Phone:
EMail: chris.newman@oracle.com

Sean Shen (editor)
CNNIC
No.4 South 4th Zhongguancun Street
Beijing, 100190
China

Phone: +86 10-58813038
EMail: shenshuo@cnnic.cn

Email Address Internationalization (EAI)
Internet-Draft
Intended status: Standards Track
Expires: April 25, 2013

K. Fujiwara
JPRS

Oct 22, 2012

Post-delivery Message Downgrading for Internationalized Email Messages
draft-ietf-eai-popimap-downgrade-08.txt

Abstract

The Email Address Internationalization (SMTPUTF8) extension to SMTP allows UTF-8 characters in mail header fields. Upgraded POP and IMAP servers support internationalized Email messages. If a POP/IMAP client does not support Email Address Internationalization, POP/IMAP servers cannot deliver Internationalized Email Headers to the client and cannot remove the message. To avoid the situation, this document describes a conversion mechanism for internationalized Email messages to be in traditional message format. In the process, message elements requiring internationalized treatment are recoded or removed and receivers are able to know that they received messages containing such elements even if they cannot process the internationalized elements.

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 25, 2013.

Table of Contents

1. Introduction

1.1. Problem statement

Traditional (legacy) mail systems, which are defined by [RFC5322] and other specifications, allow only ASCII characters in mail header field values. The SMTPUTF8 extension ([RFC6530], [RFC6531] and [RFC6532]) allow raw UTF-8 in those mail header fields.

If a header field contains non-ASCII strings, POP/IMAP servers cannot deliver Internationalized Email Headers to legacy clients which does not send UTF8 command or UTF8 capability, and because they have no obvious or standardized way to explain what is going on to those clients, cannot even safely discard the message.

1.2. Possible solutions

There are four plausible approaches to the problem, with the preferred one depending on the particular circumstances and relationship among the delivery SMTP server, the mail store, the POP or IMAP server, and the users and their MUA clients:

1. If the delivery MTA has sufficient knowledge about the POP and/or IMAP servers and clients being used, the message may be rejected as undeliverable.

2. The message may be downgraded by the POP or IMAP server, in a way that preserves maximum information at the expense of some complexity, and does not create security or operational problems in the mail system.

3. Some intermediate downgrading may be applied that balances more information loss against lower complexity and greater ease of implementation.

4. The POP or IMAP server may fabricate a message whose intent is to notify the client that an internationalized message is waiting but cannot be delivered until an upgraded client is available.

1.3. Approach taken in this specification

This specification describes the second of those options. It is worth noticing that, at least in the general case, none of these options preserve sufficient information to guarantee that it is possible to reply to an incoming message without loss of information, so the choice may be considered to be among "least bad" options. While this document specifies a well designed mechanism, it is only an interim solution while clients are being upgraded

[I-D.ietf-eai-rfc5721bis] [I-D.ietf-eai-5738bis].

This message downgrading mechanism converts mail header fields to an all-ASCII representation. The POP/IMAP servers can use the downgrading mechanism and deliver the Internationalized Email message as a traditional form. Receivers can know they received some internationalized messages or some unknown or broken messages.

[RFC6532] allows UTF-8 characters to be used in mail header fields and MIME header fields. [RFC6531] allows UTF-8 characters to be used in some trace header fields. The message downgrading mechanism specified here describes the conversion method from the internationalized messages that are defined in [RFC6530], and [RFC6532] to the traditional email messages defined in [RFC5322].

This document provides a precise definition of the minimum-information-loss message downgrading process.

Downgrading consists of the following three parts:

- New header field definitions
- Email header field downgrading
- MIME header field downgrading

Email header field downgrading is described in Section 3. It generates ASCII-only header fields.

In Section 3.1.10 of this document, header fields starting with "Downgraded-" are introduced. They preserve the information that appeared in the original header fields.

The definition of MIME header fields in Internationalized Email Messages is described in [RFC6532]. MIME header field downgrading is described in Section 4.1. It generates ASCII-only MIME header fields.

Displaying downgraded messages that originally contained internationalized header fields is out of scope of this document. A POP/IMAP client which does not support UTF8 extensions as defined for POP3 [UTF8 command] and IMAP ["ENABLE UTF8=ACCEPT" command] does not know internationalized message format described in [RFC6532].

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 RFC 2119 [RFC2119].

All specialized terms used in this specification are defined in the Overview and Framework for Internationalized Email [RFC6530], in the mail message specifications [RFC5322], or in the MIME documents [RFC2045] [RFC2047] [RFC2183] [RFC2231]. The terms "U-label", "A-label" and "IDNA" are used with the definitions from [RFC5890]. The terms "ASCII address", "non-ASCII address", "SMTPUTF8", "message", "internationalized message" are used with the definitions from [RFC6530]. The term "non-ASCII string" is used with the definitions from [RFC6532].

3. Email Header Fields Downgrading

This section defines the conversion method to ASCII for each header field that may contain non-ASCII strings. Section 3.1 describes rewriting methods for each ABNF element. Section 3.2 describes rewriting methods for each header field.

3.1. Downgrading Method for Each ABNF Element

Header field downgrading is defined below for each ABNF element. Converting the header field terminates when no non-ASCII strings remain in the header field.

[RFC5322] describes ABNF elements <group>, <mailbox>, <unstructured>, <word>, <comment>, <display-name>. [RFC2045] describes ABNF element <value>. <domain> is updated to allow non-ASCII characters in Section 3.3 of [RFC6531] and Section 3.2 of [RFC6532].

3.1.1. UNSTRUCTURED Downgrading

If the header field has an <unstructured> field that contains non-ASCII strings, apply [RFC2047] encoding with charset UTF-8.

3.1.2. WORD Downgrading

If the header field has any <word> fields that contain non-ASCII strings, apply [RFC2047] encoding with charset UTF-8.

3.1.3. COMMENT Downgrading

If the header field has any <comment> fields that contain non-ASCII strings, apply [RFC2047] encoding with charset UTF-8.

3.1.4. MIME-VALUE Downgrading

If the header field has any <value> elements defined by [RFC2045] and the elements contain non-ASCII strings, encode the <value> elements according to [RFC2231] with charset UTF-8 and leave the language information empty. If the <value> element is <quoted-string> and it contains <CFWS> outside the DQUOTE, remove the <CFWS> before this conversion.

3.1.5. DISPLAY-NAME Downgrading

If the header field has any <address> (<mailbox> or <group>) elements and they have <display-name> elements that contain non-ASCII strings, encode the <display-name> elements according to [RFC2047] with charset UTF-8. DISPLAY-NAME downgrading is the same algorithm as WORD downgrading.

3.1.6. DOMAIN Downgrading

If the header field has any <domain> elements that contain U-labels, rewrite the non-ASCII domain name into ASCII domain name using A-labels as specified in IDNA [RFC5891].

3.1.7. GROUP Downgrading

<group> is defined in Section 3.4 of [RFC5322]. The <group> elements may contain <mailbox>es which contain non-ASCII addresses.

If a <group> element contains <mailbox> elements and one of <mailbox>es contains a non-ASCII <local-part>, rewrite the <group> element as

```
display-name " " ENCODED_WORD " :;"
```

where the <ENCODED_WORD> is the original <group-list> encoded according to [RFC2047].

Otherwise, the <group> element does not contain non-ASCII <local-part>. If the <group> element contain non-ASCII <mailbox>es, they contains non-ASCII domain names. Rewrite the non-ASCII domain names into ASCII domain names using A-labels as specified in IDNA [RFC5891]. Generated <mailbox>es contain ASCII addresses only.

3.1.8. MAILBOX Downgrading

If the <local-part> of the <mailbox> element does not contain non-ASCII characters, the <domain> element contains non-ASCII characters. Rewrite the non-ASCII domain name into ASCII domain name using A-labels as specified in IDNA [RFC5891].

Otherwise, the <local-part> contains non-ASCII characters. The non-ASCII <local-part> has no equivalent format for ASCII addresses. The <addr-spec> element that contains non-ASCII strings may appear in two forms as:

"<" addr-spec ">"
addr-spec

Rewrite both as:

ENCODED-WORD " :;"

where the <ENCODED-WORD> is the original <addr-spec> encoded according to [RFC2047].

3.1.9. TYPED-ADDRESS Downgrading

If the header field contains <utf-8-type-addr> and the <utf-8-type-addr> contains raw non-ASCII strings, it is in utf-8-address form. Convert it to utf-8-addr-xtext form. Those forms are described in [RFC6533]. COMMENT downgrading is also performed in this case. If the address type is unrecognized and the header field contains non-ASCII strings, then fall back to using ENCAPSULATION on the entire header field specified in Section 3.1.10.

3.1.10. ENCAPSULATION: A Last Resort

As a last resort when header fields cannot be converted as discussed in the previous section, the fields are deleted and replaced by specialized new header fields. Those fields are defined to preserve, in encoded form, as much information as possible from the header field values of the incoming message. The syntax of these new header fields is:

```
fields                    =/ downgraded

downgraded =  "Downgraded-Message-Id:"         unstructured CRLF /
              "Downgraded-Resent-Message-Id:"  unstructured CRLF /
              "Downgraded-In-Reply-To:"        unstructured CRLF /
              "Downgraded-References:"         unstructured CRLF /
              "Downgraded-Original-Recipient:" unstructured CRLF /
              "Downgraded-Final-Recipient:"    unstructured CRLF
```

Applying this procedure to "Received:" header field is prohibited. ENCAPSULATION Downgrading is allowed for "Message-ID", "In-Reply-To:", "References:", "Original-Recipient" and "Final-Recipient" header fields.

To preserve a header field in a "Downgraded-" header field:

1. Generate a new header field.

    * The field name is a concatenation of "Downgraded-" and the original field name.

    * The initial new field value is the original header field value.

2. Treat the initial new header field value as if it were unstructured, and then apply [RFC2047] encoding with charset UTF-8 as necessary so that the resulting new header field value is completely in ASCII.

3. Remove the original header field.

3.2. Downgrading Method for Each Header Field

[RFC4021] establishes a registry of header fields. This section describes the downgrading method for each header field.

If the whole mail header field does not contain non-ASCII strings, email header field downgrading is not required. Each header field's downgrading method is described below.

3.2.1. Address Header Fields That Contain <address>s

From:
Sender:
To:
Cc:
Bcc:
Reply-To:
Resent-From:
Resent-Sender:
Resent-To:
Resent-Cc:
Resent-Bcc:
Resent-Reply-To:
Return-Path:
Disposition-Notification-To:

If the header field contains non-ASCII characters, first perform COMMENT downgrading and DISPLAY-NAME downgrading as described in the corresponding subsections of Section 3.1. If the header field still contains non-ASCII characters after that, do the following two steps:

1. If the header field contains <group> elements that contain non-ASCII addresses, perform GROUP downgrading on those elements.

2. If the header field contains <mailbox> elements that contain non-ASCII addresses, perform MAILBOX downgrading on those elements.

This procedure may generate empty <group> elements in "From:", "Sender:" and "Reply-To:" header fields.
[I-D.leiba-5322upd-from-group] updates [RFC5322] to allow (empty) <group> elements in "From:", "Sender:" and "Reply-To:" header fields.

3.2.2. Downgrading Non-ASCII in Comments

Date:
Resent-Date:
MIME-Version:
Content-ID:
Content-Transfer-Encoding:
Content-Language:
Accept-Language:
Auto-Submitted:

These header fields do not contain non-ASCII strings except in comments. If the header field contains UTF-8 characters in comments, perform COMMENT downgrading.

3.2.3. Message-ID Header Fields

Message-ID:
Resent-Message-ID:
In-Reply-To:
References:

Perform ENCAPSULATION as specified in Section 3.1.10.

3.2.4. Received Header Field

Received:

If <domain> elements or <mailbox> elements contains U-labels, perform DOMAIN downgrading specified in Section 3.1.6. Comments may contain non-ASCII strings, perform COMMENT downgrading.

After the DOMAIN downgrading and the COMMENT downgrading, if the FOR clause contains a non-ASCII <local-part>, remove the "FOR" clause. If the ID clause contains a non-ASCII values, remove the "ID" clause.

3.2.5. MIME Content Header Fields

Content-Type:
Content-Disposition:

Perform MIME-VALUE downgrading and COMMENT downgrading.

3.2.6. Non-ASCII in <unstructured>

Subject:
Comments:
Content-Description:

Perform UNSTRUCTURED downgrading.

3.2.7. Non-ASCII in <phrase>

Keywords:

Perform WORD downgrading.

3.2.8. Other Header Fields

There are other header fields that contain non-ASCII strings. They are user-defined and missing from this document, or future defined header fields. They are treated as "Optional Fields" and their field values are treated as unstructured described in Section 3.6.8 of [RFC5322].

Perform UNSTRUCTURED downgrading.

If the software understands the header field’s structure and a downgrading algorithm other than UNSTRUCTURED is applicable, that software SHOULD use that algorithm; UNSTRUCTURED downgrading is used as a last resort.

Mailing list header fields (those that start in "List-") are part of this category.

4. MIME Downgrading

Both MIME Body-Part header fields and contents of a delivery status notification may contain non-ASCII characters.

4.1. MIME Body-Part Header Field Downgrading

MIME body-part header fields may contain non-ASCII strings [RFC6532]. This section defines the conversion method to ASCII-only header fields for each MIME header field that contains non-ASCII strings. Parse the message body's MIME structure at all levels and check each MIME header field to see whether it contains non-ASCII strings. If the header field contains non-ASCII strings in the header field value, the header field is a target of the MIME body-part header field's downgrading. Each MIME header field's downgrading method is described below. COMMENT downgrading, MIME-VALUE downgrading, and UNSTRUCTURED downgrading are described in Section 3.

Content-ID:
: The "Content-ID:" header field does not contain non-ASCII strings except in comments. If the header field contains UTF-8 characters in comments, perform COMMENT downgrading.

Content-Type:
Content-Disposition:
: Perform MIME-VALUE downgrading and COMMENT downgrading.

Content-Description:
: Perform UNSTRUCTURED downgrading.

4.2. Delivery Status Notification downgrading

If the message contains a delivery status notification defined at Section 6 of [RFC3461], perform the following tests and conversions.

If there are "Original-Recipient:" and "Final-Recipient:" header fields, and the header fields contain non-ASCII strings, perform TYPED-ADDRESS downgrading.

5. Security Considerations

The purpose of post-delivery message downgrading is to allow POP/IMAP servers to deliver internationalized messages to traditional POP/IMAP clients and permit the clients to display those messages. Users who receive such messages can know that they were internationalized. It does not permit receivers to read the messages in their original form and, in general, will not permit generating replies, at least without significant user intervention.

A downgraded message's header fields contain ASCII characters only. But they still contain MIME-encapsulated header fields that contain non-ASCII strings. Furthermore, the body part may contain UTF-8 characters. Implementations parsing Internet messages need to accept

UTF-8 body parts and UTF-8 header fields that are MIME-encoded. Thus, this document inherits the security considerations of MIME-encoded header fields ([RFC2047] and [RFC3629]).

Rewriting header fields increases the opportunities for undetected spoofing by malicious senders. However, the rewritten header field values are preserved in equivalent MIME form or in newly defined header fields for which traditional MUAs have no special processing procedures.

The techniques described here invalidate methods that depend on digital signatures over any part of the message, which includes the top-level header fields and body-part header fields. Depending on the specific message being downgraded, at least the following techniques are likely to break: DomainKeys Identified Mail (DKIM), and possibly S/MIME and Pretty Good Privacy (PGP). The downgrade mechanism SHOULD NOT remove signatures even if the signatures will fail validation after downgrading. As much of the information as possible from the original message SHOULD be preserved.

While information in any email header field should usually be treated with some suspicion, current email systems commonly employ various mechanisms and protocols to make the information more trustworthy. Information in the new Downgraded-* header fields is not inspected by traditional MUAs, and may be even less trustworthy than the traditional header fields. Note that the Downgraded-* header fields could have been inserted with malicious intent (and with content unrelated to the traditional header fields), however traditional MUAs do not parse Downgraded-* header fields.

In addition, if an Authentication-Results header field [RFC5451] is present, traditional MUAs may treat that the digital signatures are valid.

See the "Security Considerations" section in [I-D.leiba-5322upd-from-group] and [RFC6530] for more discussion.

6. Implementation Notes

6.1. RFC 2047 Encoding

While [RFC2047] has a specific algorithm to deal with whitespace in adjacent encoded words, there are a number of deployed implementations that fail to implement the algorithm correctly. As a result, whitespace behavior is somewhat unpredictable in practice when multiple encoded words are used. While RFC 5322 states that implementations SHOULD limit lines to not more than 78 characters, implementations MAY choose to allow overly long encoded words in

order to work around faulty [RFC2047] implementations. Implementations that choose to do so SHOULD have an optional mechanism to limit line length to 78 characters.

7. IANA Considerations

[[RFC Editor: Please change "is asked to" to "has" (and change the verb correspondingly) when the IESG approval and IANA actions are complete.]]

[RFC5504] specified that no new header fields be registered that begin with "Downgraded-". That restriction is now lifted, and this document makes a new set of registrations, replacing the experimental fields with standard ones.

7.1. Obsolescence of Existing Downgraded-* Header Fields

The "Downgraded-*" header fields that were registered as experimental fields in [RFC5504] are no longer in use. IANA is asked to change the status from "experimental" to "obsoleted" for every name in the Permanent Message Header Field registry that begins with "Downgraded-".

7.2. Registration of New Downgraded-* Header Fields

[[RFC Editor: Please change "should be" to "have been" when the IANA actions are complete.]]

The following header fields should be registered in the Permanent Message Header Field registry, in accordance with the procedures set out in [RFC3864].

Header field name: Downgraded-Message-Id
Applicable protocol: mail
Status: standard
Author/change controller: IETF
Specification document(s): This document (Section 3.1.10)

Header field name: Downgraded-In-Reply-To
Applicable protocol: mail
Status: standard
Author/change controller: IETF
Specification document(s): This document (Section 3.1.10)

Header field name: Downgraded-References
Applicable protocol: mail
Status: standard
Author/change controller: IETF
Specification document(s): This document (Section 3.1.10)

Header field name: Downgraded-Original-Recipient
Applicable protocol: mail
Status: standard
Author/change controller: IETF
Specification document(s): This document (Section 3.1.10)

Header field name: Downgraded-Final-Recipient
Applicable protocol: mail
Status: standard
Author/change controller: IETF
Specification document(s): This document (Section 3.1.10)

8. Acknowledgements

This document draws heavily from the experimental in-transit message downgrading procedure described in RFC 5504 [RFC5504]. The contribution of the co-author of that earlier document, Y. Yoneya, are gratefully acknowledged. Significant comments and suggestions were received from John Klensin, Barry Leiba, Randall Gellens, Pete Resnick, Martin J. Durst, and other WG participants.

9. References

9.1. Normative References

[RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

[RFC2047] Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, November 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2183] Troost, R., Dorner, S., and K. Moore, "Communicating Presentation

Information in Internet Messages: The Content-Disposition Header Field", RFC 2183, August 1997.

[RFC2231] Freed, N. and K. Moore, "MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages , and Continuations", RFC 2231, November 1997.

[RFC3461] Moore, K., "Simple Mail Transfer Protocol (SMTP) Service Extension for Delivery Status Notifications (DSNs)", RFC 3461, January 2003.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

[RFC3864] Klyne, G., Nottingham, M., and J. Mogul, "Registration Procedures for Message Header Fields", BCP 90, RFC 3864, September 2004.

[RFC4021] Klyne, G. and J. Palme, "Registration of Mail and MIME Header Fields", RFC 4021, March 2005.

[RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, October 2008.

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.

[RFC5891] Klensin, J., "Internationalized Domain Names in Applications (IDNA): Protocol", RFC 5891, August 2010.

[RFC6530] Klensin, J. and Y. Ko, "Overview and Framework for Internationalized Email", RFC 6530, February 2012.

[RFC6531] Yao, J. and W. Mao, "SMTP Extension for Internationalized Email", RFC 6531, February 2012.

[RFC6532] Yang, A., Steele, S., and N. Freed, "Internationalized Email Headers", RFC 6532, February 2012.

[RFC6533] Hansen, T., Newman, C., and A. Melnikov, "Internationalized Delivery Status and Disposition Notifications", RFC 6533, February 2012.

[I-D.leiba-5322upd-from-group] Leiba, B., "Update to Internet Message Format to Allow Group Syntax in the "From:" and "Sender:" Header Fields", draft-leiba-5322upd-from-group-06 (work in progress), October 2012.

[I-D.ietf-eai-rfc5721bis] Gellens, R., Newman, C., Yao, J., and K. Fujiwara, "POP3 Support for UTF-8", draft-ietf-eai-rfc5721bis-08 (work in progress), October 2012.

[I-D.ietf-eai-5738bis] Resnick, P., Newman, C., and S. Shen, "IMAP Support for UTF-8", draft-ietf-eai-5738bis-09 (work in progress), August 2012.

9.2. Informative References

[RFC5451] Kucherawy, M., "Message Header Field for Indicating Message Authentication Status", RFC 5451, April 2009.

[RFC5504] Fujiwara, K. and Y. Yoneya, "Downgrading Mechanism for Email Address Internationalization", RFC 5504, March 2009.

Appendix A. Examples

A.1. Downgrading Example

This appendix shows an message downgrading example. Consider a received mail message where:

- The sender address is a non-ASCII address, "NON-ASCII-LOCAL@example.com". Its display-name is "DISPLAY-LOCAL".

- The "To:" header field contains two non-ASCII addresses, "NON-ASCII-REMOTE1@example.net" and "NON-ASCII-REMOTE2@example.com" Its display-names are "DISPLAY-REMOTE1" and "DISPLAY-REMOTE2".
- The "Cc:" header field contains a non-ASCII address, "NON-ASCII-REMOTE3@example.org". Its display-name is "DISPLAY-REMOTE3".
- Four display names contain non-ASCII characters.
- The Subject header field is "NON-ASCII-SUBJECT", which contains non-ASCII strings.
- The "Message-Id:" header field contains "NON-ASCII-MESSAGE_ID", which contains non-ASCII strings.
- There is an unknown header field "X-Unknown-Header" which contains non-ASCII strings.

```
Return-Path: <NON-ASCII-LOCAL@example.com>
Received: from ... by ... for <NON-ASCII-REMOTE1@example.net>
Received: from ... by ... for <NON-ASCII-REMOTE1@example.net>
From: DISPLAY-LOCAL <NON-ASCII-LOCAL@example.com>
To: DISPLAY-REMOTE1 <NON-ASCII-REMOTE1@example.net>,
    DISPLAY-REMOTE2 <NON-ASCII-REMOTE2@example.com>
Cc: DISPLAY-REMOTE3 <NON-ASCII-REMOTE3@example.org>
Subject: NON-ASCII-SUBJECT
Date: Mon, 30 Jul 2012 01:23:45 -0000
Message-Id: NON-ASCII-MESSAGE_ID
Mime-Version: 1.0
Content-Type: text/plain; charset="UTF-8"
Content-Transfer-Encoding: 8bit
X-Unknown-Header: NON-ASCII-CHARACTERS

MAIL_BODY
```

Figure 1: Received message in a mail drop

The downgraded message is shown in Figure 2. "Return-Path:", "From:", "To:" and "Cc:" header fields are rewritten. "Subject:" and "X-Unknown-Header:" header fields are encoded using [RFC2047]. "Message-Id:" header field is encapsulated as "Downgraded-Message-Id:" header field.

```
Return-Path: =?UTF-8?Q?NON-ASCII-LOCAL@example.com?= :;
Received: from ... by ...
Received: from ... by ...
From: =?UTF-8?Q?DISPLAY-LOCAL?=
      =?UTF-8?Q?NON-ASCII-LOCAL@example.com?= :;
To:   =?UTF-8?Q?DISPLAY-REMOTE1?=
      =?UTF-8?Q?NON-ASCII-REMOTE1@example.net?= :;,
      =?UTF-8?Q?DISPLAY-REMOTE2?=
      =?UTF-8?Q?NON-ASCII-REMOTE2@example.com?= :;,
Cc:   =?UTF-8?Q?DISPLAY-REMOTE3?=
      =?UTF-8?Q?NON-ASCII-REMOTE3@example.org?= :;
Subject: =?UTF-8?Q?NON-ASCII-SUBJECT?=
Date: Mon, 30 Jul 2012 01:23:45 -0000
Downgraded-Message-Id: =?UTF-8?Q?MESSAGE_ID?=
Mime-Version: 1.0
Content-Type: text/plain; charset="UTF-8"
Content-Transfer-Encoding: 8bit
X-Unknown-Header: =?UTF-8?Q?NON-ASCII-CHARACTERS?=

MAIL_BODY
```

Figure 2: Downgraded message

Appendix B. Change History

[[RFC Editor: Please remove this section prior to publication.]]

This section is used for tracking the update of this document. Will be removed after finalize.

B.1. Version 00

- Initial version
- Imported header field downgrading from RFC 5504

B.2. Version 01

- same as Version 00

B.3. Version 02

- Added updating RFC 5322 to allow <group> syntax in From: and Sender
- Added GROUP Downgrading

B.4. Version 03

- Replaced <utf8-addr-spec> with <addr-spec>
- Added updating RFC 5322 to allow <group> syntax in From: and Sender
- Added one sentence in Security considerations
- Updated IANA considerations

B.5. Version 04

- Removed "Internationalized Address removed" from GROUP and MAILBOX downgrading
- Updated "Updating RFC 5322"
- Compacted new header field definition
- Compacted security considerations
- Updated IANA considerations to remove obsoleting header fields that are registered by RFC 5504
- Added a discussion of alternate downgrading models for the POP and IMAP cases.
- Incorporated a large number of editorial changes to improve clarity.

B.6. Version 05

- Some text corrections
- Terminology change: only to use non-ASCII address, non-ASCII message, non-ASCII string and imported them from RFC 6530 and RFC 6532
- Replace "non-ASCII character" with "non-ASCII string"
- Removed 5.1.1. RECEIVED Downgrading

B.7. Version 06

- Removed "Updating RFC 5322"

- Added reference to draft-leiba-5322upd-from-group

B.8. Version 07

- Updated by WGLC comments
- Fixed Received downgrading and added to refer "RFC 6531", "RFC 5890", "RFC 5891"
- Added Domain downgrading for Received, Group and Mailbox
- Swapped section 3 and 4

B.9. Version 08

- Updated by IETF Last call and IESG comments
- Removed "Address Header Fields with Typed Addresses" and added "Delivery Status Notification downgrading" in MIME downgrading
- Added a space between display-name and ENCODED_WORD.
- Moved "ENCAPSULATION: A Last Resort" from section 4 to section 3.1.10.
- Updated address header fields downgrading
- Updated introduction, security considerations and iana considerations

Author's Address

Kazunori Fujiwara
Japan Registry Services Co., Ltd.
Chiyoda First Bldg. East 13F, 3-8-1 Nishi-Kanda
Chiyoda-ku, Tokyo 101-0065
Japan

Phone: +81 3 5215 8451
EMail: fujiwara@jprs.co.jp

Network Working Group
Internet-Draft
Obsoletes: RFC5336
(if approved)
Updates: RFC5321 and 5322
(if approved)
Intended status: Standards Track
Expires: September 8, 2011

J. Yao
W. Mao
CNNIC

March 7, 2011

SMTP Extension for Internationalized Email Address
draft-ietf-eai-rfc5336bis-08.txt

Abstract

This document specifies an SMTP extension for transport and delivery of email messages with internationalized email addresses or header information.

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 September 8, 2011.

Table of Contents

1. Introduction

The Simple Mail Transfer Protocol [RFC5321] provides a negotiation mechanism about service extension by which SMTP clients can discover SMTP server capabilities and make decisions for further processing. This document uses this mechanism and specifies an SMTP extension to permit internationalized email addresses (see Section 1.2) in the SMTP envelope, and Unicode characters encoded in UTF-8 [RFC3629] in the headers. An extended overview of the extension model for internationalized email addresses and the email header appears in [RFC4952bis], referred to as "the framework document" or just as "framework" elsewhere in this specification.

[[anchor1: Note in Draft and to RFC Editor: The keyword represented in this document by "UTF8SMTPbis" (and in the XML source byUTF8SMTPbis) is a placeholder. The actual keyword will be assigned when the standards track SMTP extension in this series [RFC5336bis-SMTP] is approved for publication and should be substituted here. This paragraph should be treated as normative reference to that SMTP extension draft, creating a reference hold until it is approved by the IESG. This paragraph should be removed before RFC publication.]]

1.1. Role of This Specification

The framework document [RFC4952bis] specifies the requirements for, and describes components of, full internationalization of electronic mail. A thorough understanding of the information in that document and in the base Internet email specifications [RFC5321] [RFC5322] is necessary to understand and implement this specification.

This document specifies an element of the email internationalization work, specifically the definition of an SMTP extension for internationalized email address transport delivery.

1.2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

The terms "UTF-8 string" or "UTF-8 character" are used to refer to Unicode characters encoded in UTF-8. All other specialized terms used in this specification are defined in the framework document or in the base Internet email specifications. In particular, the terms "ASCII address", "internationalized email address", "non-ASCII address", "UTF8SMTPbis", "internationalized message", and "message" are used in this document according to the definitions in the framework document.

Non-ASCII characters or strings referred in this document MUST be expressed in UTF-8, a standard Unicode Encoding Form.

This specification uses Augmented BNF (ABNF) rules [RFC5234], with some modifications. The modified rules are defined in this specification. When a new rule has a name starting with "u", it is a small modification to an older rule. Rules that are undefined here can be found from [RFC5234] or [RFC5321] or [RFC5322] under the same names.

1.3. Updates to Other Specifications

This specification modifies RFC 5321 by permitting internationalized email address in the envelop. It also updates some syntax rules defined in RFC 5321. It modifies RFC 5322 by permitting data formats defined in [RFC5335bis]. It does not modify the 8BITMIME specification [RFC6152] in any way, but it does require that the 8BITMIME extension be announced by the EAI-aware SMTP server and used with "BODY=8BMITMIME".

2. Overview of Operation

This specification describes an optional extension to the email transport mechanism that permits non-ASCII characters in both the envelope and header fields of messages, which are encoded in UTF-8. The extension is identified with the token "UTF8SMTPbis".

The EAI UTF-8 header specification [RFC5335bis] provides the details of email header features enabled by this extension

3. Mail Transport-Level Protocol

3.1. Framework for the Internationalization Extension

The following service extension is defined:

1. The name of the SMTP service extension is "Email Address Internationalization".
2. The EHLO keyword value associated with this extension is "UTF8SMTPbis".
3. No parameter values are defined for this EHLO keyword value. In order to permit future (although unanticipated) extensions, the EHLO response MUST NOT contain any parameters for this keyword. The EAI-aware SMTP client MUST reject any parameters if they appear for this keyword; that is, the EAI-aware SMTP client MUST behave as if the parameters do not appear. If an SMTP server includes UTF8SMTPbis in its EHLO response, it MUST be fully

compliant with this version of this specification.

4. One OPTIONAL parameter "UTF8SMTPbis" is added to the MAIL command. The parameter has no value. If this parameter is set in the MAIL command, it indicates that the SMTP client is EAI-aware and asserts that the envelop includes the non-ASCII address or the message being sent is internationalized message or the message being sent needs the UTF8SMTPbis support.
5. The maximum length of a MAIL command line is increased by 12 characters by the possible addition of the UTF8SMTPbis parameter. [[anchor6: RFC Editor: the number '12' will be replaced by the new number (1 space + length of the new keyword supposed to replace "UTF8SMTPbis").]]
6. One OPTIONAL parameter "UTF8SMTPbis" is added to the VRFY and EXPN commands. The parameter UTF8SMTPbis has no value. The parameter indicates that the SMTP client can accept Unicode characters in UTF-8 encoding in replies from the VRFY and EXPN commands.
7. No additional SMTP verbs are defined by this extension.
8. Servers offering this extension MUST provide support for, and announce, the 8BITMIME extension [RFC6152].
9. The reverse-path and forward-path of the SMTP MAIL and RCPT commands are extended to allow Unicode characters encoded in UTF-8 in mailbox names (addresses).
10. The mail message body is extended as specified in [RFC5335bis].
11. The UTF8SMTPbis extension is valid on the submission port [RFC4409], and can be used with LMTP [RFC2033].

3.2. The UTF8SMTPbis Extension

An SMTP server that announces this UTF8SMTPbis extension MUST be prepared to accept a UTF-8 string [RFC3629] in any position in which RFC 5321 specifies that a <mailbox> can appear. Although the characters in the <local-part> are permitted to contain non-ASCII characters, actual parsing of the <local-part>, and the delimiters used, are unchanged from the base email specification [RFC5321]. Any domain names to be looked up in the DNS MUST allow for [RFC5890] behavior. When doing lookups, the EAI-aware SMTP server MUST either use a Unicode aware DNS library, or transform it to A-label defined in [RFC5890].

An SMTP client that receives the UTF8SMTPbis extension keyword in response to the EHLO command MAY transmit mailbox names within SMTP commands as internationalized strings in UTF-8 form. It MAY send a UTF-8 header [RFC5335bis] (which may also include mailbox names in UTF-8). It MAY transmit the domain parts of mailbox names within SMTP commands or the message header as A-labels or U-labels [RFC5890]. The presence of the UTF8SMTPbis extension does not change RFC 5321 server relaying behaviors.

If the UTF8SMTPbis SMTP extension is not offered by the SMTP server, the EAI-aware SMTP client MUST NOT transmit an internationalized email address and MUST NOT transmit a mail message containing internationalized mail headers as described in [RFC5335bis] at any level within its MIME structure [RFC2045] and [RFC2047]. (For this paragraph, the internationalized domain name in the form of A-labels as specified in IDNA definitions [RFC5890] is not considered to be "internationalized".) Instead, if an EAI-aware SMTP client (EAI-aware SMTP sender) attempts to transfer an internationalized message and encounters an SMTP server that does not support the extension, it MUST make one of the following three choices and the priority order is 1, 2 and 3.

1. It MAY either reject the message during the SMTP transaction or accept the message and then generate and transmit a notification of non-deliverability. Such notification MUST be done as specified in RFC 5321 [RFC5321], RFC 3464 [RFC3464], and the EAI delivery status notification (DSN) specification [RFC5337bis].
2. If and only if the EAI-aware SMTP client (sender) is a Message Submission Agent ("MSA") [RFC4409] [RFC5598], it MAY rewrite the envelope, headers, or message material to make them entirely ASCII [ASCII] and consistent with the provisions of RFC 5321 [RFC5321] and RFC 5322 [RFC5322].
3. It MAY find an alternate route to the destination that permits UTF8SMTPbis. That route MAY be discovered by trying alternate Mail eXchanger (MX) hosts (using preference rules as specified in RFC 5321) or using other means available to the EAI-aware SMTP client.

This document applies only when an EAI-aware SMTP client is trying to send an internationalized message to an EAI-aware SMTP server. For all other cases, and for addresses and messages that do not require an UTF8SMTPbis extension, EAI-aware SMTP clients and servers do not change the behavior specified in [RFC5321].

An EAI-aware MUA/MSA sending to a legacy SMTP server [RFC5321] and [RFC5322] MAY convert an ASCII@U-label [RFC5890] address into the format of ASCII@A-label [RFC5890] if the email address is in the format of ASCII@U-label.

3.3. Extended Mailbox Address Syntax

RFC 5321, Section 4.1.2, defines the syntax of a <mailbox> entirely in terms of ASCII characters.

The key changes made by this specification include:

- Change the definition of <Domain> to permit both the RFC 5321 definition and a UTF-8 string representing a DNS label that is conformed with IDNA definitions [RFC5890].
- Change the definition of <Local-part> to permit both the RFC 5321 definition and a UTF-8 string. That string MUST NOT contain any of the ASCII characters (either graphics or controls) that are not permitted in <atext>; it is otherwise unrestricted.

According to the description above, the syntax of an internationalized email mailbox name (address) is defined in ABNF [RFC5234] as follows.

```
   uMailbox = uLocal-part "@" ( uDomain / address-literal )
   ; Replace Mailbox in RFC 5321, Section 4.1.2

   address-literal = <Defined in Section 4.1.2 of RFC 5321>

    uLocal-part = uDot-string / uQuoted-string
   ; MAY be case-sensitive
   ; Replace Local-part in RFC 5321, Section 4.1.2

    uDot-string = uAtom *("." uAtom)
   ; Replace Dot-string in RFC 5321, Section 4.1.2

    uAtom = 1*ucharacter
   ; Replace Atom in RFC 5321, Section 4.1.2

    ucharacter = atext / UTF8-non-ascii

    atext = <Defined in Section 3.2.3 of RFC 5322>
   ; Same definition with atext in RFC 5321, Section 4.1.2

    uQuoted-string = DQUOTE *uQcontentSMTP DQUOTE
   ; Replace Quoted-string in RFC 5321, Section 4.1.2

   DQUOTE = <Defined in appendix B.1 of RFC 5234>

    uQcontentSMTP = qtextSMTP / quoted-pairSMTP / UTF8-non-ascii

    qtextSMTP = <Defined in Section 4.1.2 of RFC 5321>

   quoted-pairSMTP = <Defined in Section 4.1.2 of RFC 5321>

    uDomain = sub-udomain *("." sub-udomain)
   ; Replace Domain in RFC 5321, Section 4.1.2

    sub-udomain = uLet-dig [uLdh-str]
   ; Replace sub-domain in RFC 5321, Section 4.1.2

    uLet-dig = Let-dig / UTF8-non-ascii

   Let-dig = <Defined in Section 4.1.2 of RFC 5321>

    uLdh-str = *( ALPHA / DIGIT / "-" / UTF8-non-ascii) uLet-dig
   ; Replace Ldh-str in RFC 5321, Section 4.1.2

   UTF8-non-ascii = <Defined in Section 4.1 of RFC5335bis>
```

3.4. MAIL Command Parameter Usage

If the envelope or message being sent requires the capabilities of the UTF8SMTPbis extension, the SMTP client MUST supply the UTF8SMTPbis parameter with the MAIL command. If this parameter is provided, it MUST have no value. If the SMTP client is aware that neither the envelope nor the message being sent requires any of the UTF8SMTPbis extension capabilities, it SHOULD NOT supply the UTF8SMTPbis parameter with the MAIL command.

Because there is no guarantee that a next-hop SMTP server will support the UTF8SMTPbis extension, use of the UTF8SMTPbis extension always carries a risk of transmission failure. In fact, during the early stages of deployment for the UTF8SMTPbis extension, the risk will be quite high. Hence there is a distinct near-term advantage for ASCII-only messages to be sent without using this extension. The long-term advantage of casting ASCII [ASCII] characters(0x7f and below) as UTF-8 form is that it permits pure-Unicode environments.

This specification does not require that the EAI-aware SMTP client inspect the message or otherwise go to extraordinary lengths to assure itself whether the UTF8SMTPbis extension is REQUIRED for the particular message.

3.5. Non-ASCII addresses and Reply-codes

An EAI-aware SMTP client MUST only send an internationalized message to an SMTP server that supports UTF8SMTPbis. If the SMTP server does not support this option, then the EAI-aware SMTP client has three choices according to section 3.2 of this specification and MAY choose to reject the internationalized message.

The three-digit Reply-codes used in this section are based on their meanings as defined in RFC 5321.

When messages are rejected because the RCPT command requires an ASCII address, the reply-code 553 is returned with the meaning "mailbox name not allowed". When messages are rejected because the MAIL command requires an ASCII address, the reply-code 550 is returned with the meaning "mailbox unavailable". When the EAI-aware SMTP server supports enhanced mail system status codes [RFC3463], reply-code "X.6.7" [RFC5248] is used, meaning that "non-ASCII addresses not permitted for that sender/recipient".

When messages are rejected for other reasons, the server SHOULD follow the model of the base email specifications [RFC5321]; this extension does not change those circumstances or reply messages.

If the reply-code is issued after the final "." of the DATA command, the reply-code "554" is used with the meaning "Transaction failed". When the EAI-aware SMTP server supports enhanced mail system status codes [RFC3463], reply-code "X.6.9" [RFC5248] is used, meaning that "UTF-8 header message can not be transmitted to one or more recipients, so the message MUST be rejected".

3.6. Body Parts and SMTP Extensions

The MAIL command parameter UTF8SMTPbis asserts that a message is an internationalized message or the message being sent needs the UTF8SMTPbis support. The message being sent via the MAIL command with the UTF8SMTPbis parameter has still a chance of that the message transmitted is not an internationalized message. An EAI-aware SMTP client or server that requires accurate knowledge of whether a message is internationalized needs to parse all message header fields and MIME header fields [RFC2045] and [RFC2047] in the message body. However, this specification does not require that the SMTP client or server inspects the message.

While this specification requires that EAI-aware SMTP servers support the 8BITMIME extension [RFC6152] to ensure that servers have adequate handling capability for 8-bit data and to avoid a number of complex encoding problems, the use of internationalized email addresses obviously does not require non-ASCII body parts in the MIME message in RFC 2045 and RFC 2047. The UTF8SMTPbis extension MAY be used with the BODY=8BITMIME parameter [RFC6152] if that is appropriate given the body content or, with the BODY=BINARYMIME parameter, if the SMTP server advertises BINARYMIME [RFC3030] and that is appropriate.

3.7. Additional ESMTP Changes and Clarifications

The information carried in the mail transport process involves addresses ("mailboxes") and domain names in various contexts in addition to the MAIL and RCPT commands and extended alternatives to them. In general, the rule is that, when RFC 5321 specifies a mailbox, this SMTP extension requires UTF-8 form to be used for the entire string; when RFC 5321 specifies a domain name, the name SHOULD be in the form of A-label if this domain name is an internationalized domain name[RFC5890].

The following subsections list and discuss all of the relevant cases.

3.7.1. The Initial SMTP Exchange

When an SMTP connection is opened, the SMTP server sends a "greeting" response consisting of the 220 reply-code and some information. The SMTP client then sends the EHLO command. Since the SMTP client

cannot know whether the SMTP server supports UTF8SMTPbis until after it receives the response from EHLO, the EAI-aware SMTP client MUST send only ASCII (LDH label or A-label [RFC5890]) domains in the EHLO command and that, if the EAI-aware SMTP server provides domain names in the EHLO response, they MUST be in the form of LDH labels or A-labels.

3.7.2. Mail eXchangers

If multiple DNS MX records are used to specify multiple servers for a domain in section 5 of [RFC5321], it is strongly advised that all or none of them SHOULD support the UTF8SMTPbis extension. Otherwise, surprising rejections can happen during temporary or permanent failures, which users might perceive as serious reliability issues. In order to avoid the possible surprising rejections, the EAI-aware email system MAY also implement the advice in EAI addresses advice document [EAI addresses] and EAI deployment advice document [EAI Deployment].

3.7.3. Trace Information

For the trace information [RFC5321], this memo updates the time stamp line and the return path line [RFC5321] formally defined as follows:

```
   uReturn-path-line = "Return-Path:" FWS uReverse-path <CRLF>
     ; Replaces Return-path-line in Section 4.4 of RFC 5321

   uReverse-path = uPath / "<>"
     ; Replace Reverse-path in RFC 5321, section 4.1.2

   uPath = "<" [ A-d-l ":" ] uMailbox ">"
     ; Replace Path in RFC 5321, section 4.1.2
     ; uMailbox is defined in section 3.3 of this document

   A-d-l = <Defined in section 4.1.2 of RFC 5321>

   uTime-stamp-line = "Received:" FWS uStamp <CRLF>
     ; Replaces Time-stamp-line in Section 4.4 of RFC 5321

   uStamp = From-domain By-domain uOpt-info [CFWS] ";" FWS date-time
     ; Replaces Stamp in Section 4.4 of RFC 5321

   From-domain = <Defined in section 4.4 of RFC 5321>

   By-domain = <Defined in section 4.4 of RFC 5321>

   date-time = <Defined in section 3.3 of RFC 5322>
       ; Same definition with date-time in Section 4.4 of RFC 5321

   uOpt-info = [Via] [With] [ID] [uFor]
         [Additional-Registered-Clauses]
     ; Replaces Opt-info in Section 4.4 of RFC 5321
     ; The protocol value for With will allow a UTF8SMTPbis value

   Via = <Defined in section 4.4 of RFC 5321>

   With = <Defined in section 4.4 of RFC 5321>

   ID = <Defined in section 4.4 of RFC 5321>

   Additional-Registered-Clauses = <Defined in section 4.4 of RFC 5321>

   uFor = CFWS "FOR" FWS ( uPath / uMailbox)
     ; Replaces For in Section 4.4 of RFC 5321
     ; uMailbox is defined in section 3.3 of this document
```

Except in the 'uFor' clause and 'uReverse-path' value where internationalized domain name with the U-label form MAY be used, internationalized domain names in Received fields MUST be transmitted in the form of A-labels. The protocol value of the WITH clause when this extension is used is one of the UTF8SMTPbis values specified in the "IANA Considerations" section of this document.

3.7.4. UTF-8 Strings in Replies

3.7.4.1. MAIL and RCPT Commands

If an SMTP client follows this specification and sends any MAIL commands containing the UTF8SMTPbis parameter or any RCPT commands containing non-ASCII addresses, the EAI-aware SMTP server is permitted to use UTF-8 characters in the email address associated with 251 and 551 reply-codes, and the SMTP client MUST be able to accept and process them. If a given MAIL command does not include the UTF8SMTPbis parameter or a given RCPT command does not include a non-ASCII envelope address, the EAI-aware SMTP server MUST NOT return a 251 or 551 response containing a non-ASCII mailbox. Instead, it MUST transform such responses into 250 or 550 responses that do not contain non-ASCII addresses.

3.7.4.2. VRFY and EXPN Commands and the UTF8SMTPbis Parameter

If the VRFY and EXPN commands are transmitted with the parameter "UTF8SMTPbis", it indicates the SMTP client can accept UTF-8 strings in replies to those commands. This parameter for the VRFY and EXPN commands SHOULD only be used after the SMTP client sees the EHLO response with the UTF8SMTPbis keyword. This allows the EAI-aware SMTP server to use UTF-8 strings in mailbox names and full names that occur in replies without concern that the SMTP client might be confused by them. An SMTP client that conforms to this specification MUST accept and correctly process replies from the VRFY and EXPN commands that contain UTF-8 strings. However, the EAI-aware SMTP server MUST NOT use UTF-8 strings in replies if the SMTP client does not specifically allow such replies by transmitting this parameter. Most replies do not require that a mailbox name be included in the returned text, and therefore UTF-8 string is not needed in them. Some replies, notably those resulting from successful execution of the VRFY and EXPN commands, do include the mailbox.

VERIFY (VRFY) and EXPAND (EXPN) command syntaxes are changed to:

```
 vrfy = "VRFY" SP uString
  [ SP "UTF8SMTPbis" ] CRLF

 expn = "EXPN" SP uString
  [ SP "UTF8SMTPbis" ] CRLF

 uString = uAtom / uQuoted-string
; uAtom and uQuoted-string are defined in
; Section 3.3 of this document.
```

The "UTF8SMTPbis" parameter does not use a value. If the reply to a

VERIFY (VRFY) or EXPAND (EXPN) command requires UTF-8 string, but the SMTP client did not use the "UTF8SMTPbis" parameter, then the EAI-aware SMTP server MUST use either the reply-code 252 or 550. Reply-code 252, defined in [RFC5321], means "Cannot VRFY user, but will accept the message and attempt the delivery". Reply-code 550, also defined in [RFC5321], means "Requested action not taken: mailbox unavailable". When the EAI-aware SMTP server supports enhanced mail system status codes [RFC3463], the enhanced reply-code as specified below is used. Using the "UTF8SMTPbis" parameter with a VERIFY (VRFY) or EXPAND (EXPN) command enables UTF-8 replies for that command only.

If a normal success response (i.e., 250) is returned, the response MAY include the full name of the user and MUST include the mailbox of the user. It MUST be in either of the following forms:

```
User Name <uMailbox>
  ; uMailbox is defined in Section 3.3 of this document.
  ; User Name can contain non-ASCII characters.

uMailbox
  ; uMailbox is defined in Section 3.3 of this document.
```

If the SMTP reply requires UTF-8 strings, but UTF-8 string is not allowed in the reply, and the EAI-aware SMTP server supports enhanced mail system status codes [RFC3463], the enhanced reply-code is "X.6.8" [RFC5248], meaning "A reply containing a UTF-8 string is REQUIRED to show the mailbox name, but that form of response is not permitted by the SMTP client".

If the SMTP client does not support the UTF8SMTPbis extension, but receives a UTF-8 string in a reply, it may not be able to properly report the reply to the user, and some clients might crash. Internationalized messages in replies are only allowed in the commands under the situations described above. Under any other circumstances, UTF-8 string MUST NOT appear in the reply.

Although UTF-8 form is needed to represent email addresses in responses under the rules specified in this section, this extension does not permit the use of UTF-8 string for any other purposes. EAI-aware SMTP servers MUST NOT include non-ASCII characters in replies except in the limited cases specifically permitted in this section.

4. IANA Considerations

IANA SHOULD add a new value "UTF8SMTPbis" to the SMTP Service Extension subregistry of the Mail Parameters registry, according to

the following data:

Keywords	Description	Reference
UTF8SMTPbis	Internationalized email address	[RFCXXXX]

This document updates the values to the SMTP Enhanced Status Code subregistry of the Mail Parameters registry, following the guidance in Sections 3.5 and 3.7.4.2 of this document, and being based on [RFC5248]. The registration data is as follows:

Code: X.6.7
Sample Text: non-ASCII addresses not permitted for that sender/recipient
Associated basic status code: 550, 553
Description: This indicates the reception of a MAIL or RCPT command that non-ASCII addresses are not permitted
Defined: RFC XXXX (Standard track)
Submitter: Jiankang YAO
Change controller: ima@ietf.org

Code: X.6.8
Sample Text: UTF-8 string reply is required, but not permitted by the SMTP client
Associated basic status code: 252, 550, 553
Description: This indicates that a reply containing a UTF-8 string is required to show the mailbox name, but that form of response is not permitted by the SMTP client.
Defined: RFC XXXX (Standard track)
Submitter: Jiankang YAO
Change controller: ima@ietf.org

Code: X.6.9
Sample Text: UTF-8 header message can not be transferred to one or more recipient so the message must be rejected
Associated basic status code: 550
Description: This indicates that transaction failed after the final "." of the DATA command.
Defined: RFC XXXX (Standard track)
Submitter: Jiankang YAO
Change controller: ima@ietf.org

```
Code:       X.6.10
Description:    This is a duplicate of X.6.8 and
      SHOULD be deprecated for further use.
```

The following entries SHOULD be updated or added in the "Mail Transmission Types" registry under the Mail Parameters registry.

WITH protocol types	Description	Reference
UTF8SMTP	ESMTP with UTF8SMTP	[RFCXXXX]
UTF8SMTPA	ESMTP with UTF8SMTP and SMTP AUTH	[RFC4954] [RFCXXXX]
UTF8SMTPS	ESMTP with UTF8SMTP and STARTTLS	[RFC3207] [RFCXXXX]
UTF8SMTPSA	ESMTP with UTF8SMTP and both STARTTLS and SMTP AUTH	[RFC3207] [RFC4954] [RFCXXXX]
UTF8LMTP	LMTP with UTF8SMTP	[RFCXXXX]
UTF8LMTPA	LMTP with UTF8SMTP and SMTP AUTH	[RFC4954] [RFCXXXX]
UTF8LMTPS	LMTP with UTF8SMTP and STARTTLS	[RFC3207] [RFCXXXX]
UTF8LMTPSA	LMTP with UTF8SMTP and both STARTTLS and LMTP AUTH	[RFC3207] [RFC4954] [RFCXXXX]

5. Security Considerations

The extended security considerations discussion in the framework document [RFC4952bis] will be applied here.

More security considerations are discussed below:

Beyond the use inside the email global system (in SMTP envelopes and message headers), internationalized email addresses will also show up inside other cases, in particular:

- the logging systems of SMTP transactions and other logs to monitor the email systems;
- the trouble ticket systems used by Security Teams to manage security incidents, when an email address is involved;

This will likely require extending support for full UTF-8 also into

these systems, in order to avoid problems, which could cause also important loss of data, or require to provide an adequate mechanism to map non-ASCII strings into them.

Another security aspect to be considered is related to the ability by security team members to quickly understand, read and identify email addresses from the logs, when they are tracking an incident. Mechanims to automatically and quickly provide the origin or ownership of an internationalized email address SHALL be implemented for use also by log readers which cannot read easily non-ASCII information.

The SMTP commands VRFY and EXPN are sometimes used in SMTP transactions where there is no message to transfer (by tools used to take automated actions in case potential spam messages are identified). RFC 5321 section 3.5 and 7.3 give some detailed description of use and possible behaviours. Implementation of internationalized addrsses can affect also logs and actions by these tools.

6. Acknowledgements

This document revised the [RFC5336]document based on the EAI WG's discussion result. Many EAI WG members did some tests and implementations to move this document to the Standard Track document. Significant comments and suggestions were received from Xiaodong LEE, Nai-Wen Hsu, Yangwoo KO, Yoshiro YONEYA, and other members of the JET team and were incorporated into the specification. Additional important comments and suggestions, and often specific text, were contributed by many members of the WG and design team. Those contributions include material from John C Klensin, Charles Lindsey, Dave Crocker, Harald Tveit Alvestrand, Marcos Sanz, Chris Newman, Martin Duerst, Edmon Chung, Tony Finch, Kari Hurtta, Randall Gellens, Frank Ellermann, Alexey Melnikov, Pete Resnick, S. Moonesamy, Soobok Lee, Shawn Steele, Alfred Hoenes, Miguel Garcia, Magnus Westerlund, and Lars Eggert. Of course, none of the individuals are necessarily responsible for the combination of ideas represented here.

7. Change History

[[anchor14: RFC Editor: Please remove this section.]]

7.1. draft-yao-eai-rfc5336bis: Version 00

Applied errata suggested by Alfred Hoenes.

7.2. draft-ietf-eai-rfc5336bis: Version 00

Applied the changes suggested by the EAI new charter.

7.3. draft-ietf-eai-rfc5336bis: Version 01

Applied the changes suggested by 78 IETF EAI meeting.

7.4. draft-ietf-eai-rfc5336bis: Version 02

remove the appendix since rfc4952bis has added this material

improve the text

remove the text about no body parameter

7.5. draft-ietf-eai-rfc5336bis: Version 03

improve the text

7.6. draft-ietf-eai-rfc5336bis: Version 04

update the abstract

improve the text

7.7. draft-ietf-eai-rfc5336bis: Version 05

improve the text based on AD and Co-chairs

7.8. draft-ietf-eai-rfc5336bis: Version 06

update the iana consideration

7.9. draft-ietf-eai-rfc5336bis: Version 07

improve the iana consideration

7.10. draft-ietf-eai-rfc5336bis: Version 08

improve the texts

add the mail parameter

add the new section about mail command parameter usage

update the security consideration

8. References

8.1. Normative References

[ASCII] American National Standards Institute (formerly United States of America Standards Institute), "USA Code for Information Interchange", ANSI X3.4-1968, 1968.

[RFC2033] Myers, J., "Local Mail Transfer Protocol", RFC 2033, October 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3463] Vaudreuil, G., "Enhanced Mail System Status Codes", RFC 3463, January 2003.

[RFC3464] Moore, K. and G. Vaudreuil, "An Extensible Message Format for Delivery Status Notifications", RFC 3464, January 2003.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", RFC 3629, November 2003.

[RFC4409] Gellens, R. and J. Klensin, "Message Submission for Mail", RFC 4409, April 2006.

[RFC4952bis] Klensin, J. and Y. Ko, "Overview and Framework for Internationalized Email", I-D rfc4952bis, September 2010.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC5248] Hansen , T. and J. Klensin, "A Registry for SMTP Enhanced Mail System Status Codes", RFC 5248, June 2008.

[RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, October 2008.

[RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, October 2008.

[RFC5335bis] Abel, Y. and S. Steel, "Internationalized Email Headers", I-D rfc5335bis, March 2011.

[RFC5337bis]

Hansen, T., Ed., Newman, C., and A. Melnikov, Ed., "Internationalized Delivery Status and Disposition Notifications", I-D 5337bis, October 2010.

[RFC5890] Klensin, J., "Internationalizing Domain Names in Applications (IDNA definitions)", RFC 5890, June 2010.

[RFC5891] Klensin, J., "Internationalized Domain Names in Applications (IDNA): Protocol", RFC 5891, August 2010.

[RFC6152] Klensin, J., Freed, N., Rose, M., and D. Crocker, "SMTP Service Extension for 8-bit MIME Transport", STD 71, RFC 6152, March 2011.

8.2. Informative References

[EAI Deployment] Yao, J., Lee, X., and S. Steel, "Advice for EAI deployment", draft eai-deployment, December 2010.

[EAI addresses] Steel, S., Yao, J., and Mark. Davis, "Advice for non-ASCII & ASCII addresses", draft eai-address-advice, December 2010.

[RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

[RFC2047] Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, November 1996.

[RFC3030] Vaudreuil, G., "SMTP Service Extensions for Transmission of Large and Binary MIME Messages", RFC 3030, December 2000.

[RFC3207] Hoffman, P., "SMTP Service Extension for Secure SMTP over Transport Layer Security", RFC 3207, February 2002.

[RFC4954] Siemborski, R. and A. Melnikov, "SMTP Service Extension for Authentication", RFC 4954, July 2007.

[RFC5336] Yao, J. and W. Mao, "SMTP Extension for Internationalized Email Addresses", RFC 5336, September 2008.

[RFC5598] Crocker, D., "Internet Mail Architecture", RFC 5598, July 2009.

Authors' Addresses

Jiankang YAO
CNNIC
No.4 South 4th Street, Zhongguancun
Beijing

Phone: +86 10 58813007
Email: yaojk@cnnic.cn

Wei MAO
CNNIC
No.4 South 4th Street, Zhongguancun
Beijing

Phone: +86 10 58812230
Email: maowei_ietf@cnnic.cn

Network Working Group
Internet-Draft
Obsoletes: RFC5721
(if approved)
Updates: RFC1939
(if approved)
Intended status: Standards Track
Expires: March 14, 2011

R. Gellens
QUALCOMM Incorporated
C. Newman
Oracle
Jiankang. Yao
CNNIC
Kazunori. Fujiwara
JPRS

September 28, 2010

POP3 Support for UTF-8
draft-ietf-eai-rfc5721bis-00.txt

Abstract

This specification extends the Post Office Protocol version 3 (POP3) to support un-encoded international characters in user names, passwords, mail addresses, message headers, and protocol-level textual error strings.

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 March 14, 2011.

Copyright Notice

Table of Contents

1. Introduction

This document forms part of the Email Address Internationalization (EAI) protocols described in the EAI Framework document [I-D.ietf-eai-frmwrk-4952bis]. As part of the overall EAI work, email messages may be transmitted and delivered containing un-encoded UTF-8 characters, and mail drops that are accessed using POP3 [RFC1939] might natively store UTF-8.

This specification extends POP3 [RFC1939] using the POP3 extension mechanism [RFC2449] to permit un-encoded UTF-8 [RFC3629] in headers, as described in "Internationalized Email Headers" [I-D.ietf-eai-rfc5335bis]. It also adds a mechanism to support login names and passwords outside the ASCII character set, and a mechanism to support UTF-8 protocol-level error strings in a language appropriate for the user.

Within this specification, the term "down-conversion" refers to the process of modifying a message containing UTF-8 headers [I-D.ietf-eai-rfc5335bis] or body parts with 8bit content-transfer-encoding, as defined in MIME Section 2.8 [RFC2045], into conforming 7-bit Internet Message Format [RFC5322] with message header extensions for non-ASCII text [RFC2047] and other 7-bit encodings. Down-conversion is specified by "Message-Downgrading for Email Address Internationalization" [message-downgrade].

1.1. Conventions Used in This Document

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 "Key words for use in RFCs to Indicate Requirement Levels" [RFC2119].

The formal syntax uses the Augmented Backus-Naur Form (ABNF) [RFC5234] notation, including the core rules defined in Appendix B of RFC 5234.

In examples, "C:" and "S:" indicate lines sent by the client and server, respectively. If a single "C:" or "S:" label applies to multiple lines, then the line breaks between those lines are for editorial clarity only and are not part of the actual protocol exchange.

Note that examples always use 7-bit ASCII characters due to limitations of this document format; in particular, some examples for the "LANG" command may appear silly as a result.

2. LANG Capability

Per "POP3 Extension Mechanism" [RFC2449], this document adds a new capability response tag to indicate support for a new command: LANG. The capability tag and new command are described below.

CAPA tag:
LANG

Arguments with CAPA tag:
none

Added Commands:
LANG

Standard commands affected:
All

Announced states / possible differences:
both / no

Commands valid in states:
AUTHENTICATION, TRANSACTION

Specification reference:
this document

Discussion:

POP3 allows most +OK and -ERR server responses to include human-readable text that, in some cases, might be presented to the user. But that text is limited to ASCII by the POP3 specification [RFC1939]. The LANG capability and command permit a POP3 client to negotiate which language the server should use when sending human-readable text.

A server that advertises the LANG extension MUST use the language "i-default" as described in [RFC2277] as its default language until another supported language is negotiated by the client. A server MUST include "i-default" as one of its supported languages.

The LANG command requests that human-readable text included in all subsequent +OK and -ERR responses be localized to a language matching the language range argument (the "Basic Language Range" as described by [RFC4647]). If the command succeeds, the server returns a +OK response followed by a single space, the exact language tag selected, another space, and the rest of the line is human-readable text in the appropriate language. This and subsequent protocol-level human-

readable text is encoded in the UTF-8 charset.

If the command fails, the server returns an -ERR response and subsequent human-readable response text continues to use the language that was previously active (typically i-default).

The special "*" language range argument indicates a request to use a language designated as preferred by the server administrator. The preferred language MAY vary based on the currently active user.

If no argument is given and the POP3 server issues a positive response, then the response given is multi-line. After the initial +OK, for each language tag the server supports, the POP3 server responds with a line for that language. This line is called a "language listing".

In order to simplify parsing, all POP3 servers are required to use a certain format for language listings. A language listing consists of the language tag [RFC5646] of the message, optionally followed by a single space and a human-readable description of the language in the language itself, using the UTF-8 charset.

Examples:

```
< Note that some examples do not include the correct character
accents due to limitations of this document format. >

< The server defaults to using English i-default responses until
the client explicitly changes the language. >

C: USER karen
S: +OK Hello, karen
C: PASS password
S: +OK karen's maildrop contains 2 messages (320 octets)

< Client requests deprecated MUL language. Server replies
with -ERR response. >

C: LANG MUL
S: -ERR invalid language MUL

< A LANG command with no parameters is a request for
a language listing. >

C: LANG
S: +OK Language listing follows:
S: en English
S: en-boont English Boontling dialect
```

```
      S: de Deutsch
      S: it Italiano
      S: es Espanol
      S: sv Svenska
      S: i-default Default language
      S: .

      < A request for a language listing might fail. >

      C: LANG
      S: -ERR Server is unable to list languages

      < Once the client changes the language, all responses will be in
      that language, starting with the response to the LANG command. >

      C: LANG es
      S: +OK es Idioma cambiado

      < If a server does not support the requested primary language,
      responses will continue to be returned in the current language
      the server is using. >

      C: LANG uga
      S: -ERR es Idioma <<UGA>> no es conocido

      C: LANG sv
      S: +OK sv Kommandot "LANG" lyckades

      C: LANG *
      S: +OK es Idioma cambiado
```

3. UTF8 Capability

Per "POP3 Extension Mechanism" [RFC2449], this document adds a new capability response tag to indicate support for new server functionality, including a new command: UTF8. The capability tag and new command and functionality are described below.

CAPA tag:
   UTF8

Arguments with CAPA tag:
   USER

Added Commands:
   UTF8

Standard commands affected:
USER, PASS, APOP, LIST, TOP, RETR

Announced states / possible differences:
both / no

Commands valid in states:
AUTHORIZATION

Specification reference:
this document

Discussion:

This capability adds the "UTF8" command to POP3. The UTF8 command switches the session from ASCII to UTF-8 mode.

3.1. The UTF8 Command

The UTF8 command enables UTF-8 mode. The UTF8 command has no parameters.

Maildrops can natively store UTF-8 or be limited to ASCII. UTF-8 mode has no effect on messages in an ASCII-only maildrop. Messages in native UTF-8 maildrops can be ASCII or UTF-8 using internationalized headers [I-D.ietf-eai-rfc5335bis] and/or 8bit content-transfer-encoding, as defined in MIME Section 2.8 [RFC2045]. In UTF-8 mode, both UTF-8 and ASCII messages are sent to the client as-is (without conversion). When not in UTF-8 mode, UTF-8 messages in a native UTF-8 maildrop MUST NOT be sent to the client as-is. UTF-8 messages in a native UTF-8 maildrop MUST be down-converted (downgraded) to comply with unextended POP and Internet Mail Format without UTF-8 mode support.

Note that even in UTF-8 mode, MIME binary content-transfer-encoding is still not permitted.

The octet count (size) of a message reported in a response to the LIST command SHOULD match the actual number of octets sent in a RETR response (not counting byte-stuffing). Sizes reported elsewhere, such as in STAT responses and non-standardized, free-form text in positive status indicators (following "+OK") need not be accurate, but it is preferable if they are.

Mail stores are either ASCII or native UTF-8, and clients either issue the UTF8 command or not. The message needs converting only when it is native UTF-8 and the client has not issued the UTF8 command, in which case the server must down-convert it. The down-

converted message may be larger. The server may choose various strategies regarding down-conversion, which include when to down-convert, whether to cache or store the down-converted form of a message (and if so, for how long), and whether to calculate or retain the size of a down-converted message independently of the down-converted content. If the server does not have immediate access to the accurate down-converted size, it may be faster to estimate rather than calculate it. Servers are expected to normally follow the RFC 1939 [RFC1939] text on using the "exact size" in a scan listing, but there may be situations with maildrops containing very large numbers of messages in which this might be a problem. If the server does estimate, reporting a scan listing size smaller than what it turns out to be could be a problem for some clients. In summary, it is better for servers to report accurate sizes, but if this is not possible, high guesses are better than small ones. Some POP servers include the message size in the non-standardized text response following "+OK" (the 'text' production of RFC 2449 [RFC2449]), in a RETR or TOP response (possibly because some examples in POP3 [RFC1939] do so). There has been at least one known case of a client relying on this to know when it had received all of the message rather than following the POP3 [RFC1939] rule of looking for a line consisting of a termination octet (".") and a CRLF pair. While any such client is non-compliant, if a server does include the size in such text, it is better if it is accurate.

Clients MUST NOT issue the STLS command [RFC2595] after issuing UTF8; servers MAY (but are not required to) enforce this by rejecting with an "-ERR" response an STLS command issued subsequent to a successful UTF8 command. (Because this is a protocol error as opposed to a failure based on conditions, an extended response code [RFC2449] is not specified.)

3.2. USER Argument to UTF8 Capability

If the USER argument is included with this capability, it indicates that the server accepts UTF-8 user names and passwords.

Servers that include the USER argument in the UTF8 capability response SHOULD apply SASLprep [RFC4013] to the arguments of the USER and PASS commands.

A client or server that supports APOP and permits UTF-8 in user names or passwords MUST apply SASLprep [RFC4013] to the user name and password used to compute the APOP digest.

When applying SASLprep [RFC4013], servers MUST reject UTF-8 user names or passwords that contain a Unicode character listed in Section 2.3 of SASLprep [RFC4013]. When applying SASLprep to the USER

argument, the PASS argument, or the APOP username argument, a compliant server or client MUST treat them as a query string (i.e., unassigned Unicode code points are allowed). When applying SASLprep to the APOP password argument, a compliant server or client MUST treat them as a stored string (i.e., unassigned Unicode code points are prohibited).

The client does not need to issue the UTF8 command prior to using UTF-8 in authentication. However, clients MUST NOT use UTF-8 characters in USER, PASS, or APOP commands unless the USER argument is included in the UTF8 capability response.

The server MUST reject UTF-8 user names or passwords that fail to comply with the formal syntax in UTF-8 [RFC3629].

Use of UTF-8 characters in the AUTH command is governed by the POP3 SASL [RFC5034] mechanism.

4. Native UTF-8 Maildrops

When a POP3 server uses a native UTF-8 maildrop, it is the responsibility of the server to comply with the POP3 base specification [RFC1939] and Internet Message Format [RFC5322] when not in UTF-8 mode. Mechanisms for 7-bit downgrading to help comply with the standards are described in [message-downgrade].

5. IANA Considerations

This specification adds two new capabilities ("UTF8" and "LANG") to the POP3 capability registry [RFC2449].

6. Security Considerations

The security considerations of UTF-8 [RFC3629] and SASLprep [RFC4013] apply to this specification, particularly with respect to use of UTF-8 in user names and passwords.

The "LANG *" command might reveal the existence and preferred language of a user to an active attacker probing the system if the active language changes in response to the USER, PASS, or APOP commands prior to validating the user’s credentials. Servers MUST implement a configuration to prevent this exposure.

It is possible for a man-in-the-middle attacker to insert a LANG command in the command stream, thus making protocol-level diagnostic responses unintelligible to the user. A mechanism to integrity-protect the session, such as Transport Layer Security (TLS) [RFC2595] can be used to defeat such attacks.

Modifying server authentication code (in this case, to support UTF8 command) needs to be done with care to avoid introducing vulnerabilities (for example, in string parsing).

The UTF8 command description (Section 3.1) contains a discussion on reporting inaccurate sizes. An additional risk to doing so is that, if a client allocates buffers based on the reported size, it may overrun the buffer, crash, or have other problems if the message data is larger than reported.

7. References

7.1. Normative References

[I-D.ietf-eai-frmwrk-4952bis] Klensin, J. and Y. Ko, "Overview and Framework for Internationalized Email", draft-ietf-eai-frmwrk-4952bis-07 (work in progress), August 2010.

[I-D.ietf-eai-rfc5335bis] Yang, A. and S. Steele, "Internationalized Email Headers", draft-ietf-eai-rfc5335bis-02 (work in progress), August 2010.

[RFC1939] Myers, J. and M. Rose, "Post Office Protocol - Version 3", STD 53, RFC 1939, May 1996.

[RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

[RFC2047] Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, November 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2277] Alvestrand, H., "IETF Policy on Character Sets and Languages", BCP 18, RFC 2277, January 1998.

[RFC2449] Gellens, R., Newman, C., and L. Lundblade, "POP3 Extension Mechanism", RFC 2449, November 1998.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

[RFC4013] Zeilenga, K., "SASLprep: Stringprep Profile for User Names and Passwords", RFC 4013, February 2005.

[RFC4647] Phillips, A. and M. Davis, "Matching of Language Tags", BCP 47, RFC 4647, September 2006.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, October 2008.

[RFC5646] Phillips, A. and M. Davis, "Tags for Identifying Languages", BCP 47, RFC 5646, September 2009.

7.2. Informative References

[RFC2595] Newman, C., "Using TLS with IMAP, POP3 and ACAP", RFC 2595, June 1999.

[RFC4952] Klensin, J. and Y. Ko, "Overview and Framework for Internationalized Email", RFC 4952, July 2007.

[RFC5034] Siemborski, R. and A. Menon-Sen, "The Post Office Protocol (POP3) Simple Authentication and Security Layer (SASL) Authentication Mechanism", RFC 5034, July 2007.

[message-downgrade] Fujiwara, K. and Y. Yoneya, "Message Downgrading for Email Address Internationalization (EAI) Maildrops", draft-ietf-eai-rfc5504bis-00 (work in progress), Sep 2010.

Appendix A. Design Rationale

This non-normative section discusses the reasons behind some of the design choices in the above specification.

Due to interoperability problems with RFC 2047 and limited deployment of RFC 2231, it is hoped these 7-bit encoding mechanisms can be deprecated in the future when UTF-8 header support becomes prevalent.

USER is optional because the implementation burden of SASLprep [RFC4013] is not well understood, and mandating such support in all cases could negatively impact deployment.

While it is possible to provide useful examples for language negotiation without support for non-ASCII characters, it is difficult to provide useful examples for commands specifically designed to use the UTF-8 charset un-encoded when the document format is limited to ASCII. As a result, there are no plans to provide examples for that part of the specification as long as this remains an experimental proposal. However, implementers of this specification are encouraged to provide examples to the document authors for a future revision.

Appendix B. Acknowledgments

Thanks to John Klensin, Tony Hansen, and other EAI working group participants who provided helpful suggestions and interesting debate that improved this specification.

Authors' Addresses

Randall Gellens
QUALCOMM Incorporated
5775 Morehouse Drive
San Diego, CA 92651
US

EMail: rg+ietf@qualcomm.com

Chris Newman
Oracle
800 Royal Oaks
Monrovia, CA 91016-6347
US

EMail: chris.newman@oracle.com

Jiankang YAO
CNNIC
No.4 South 4th Street, Zhongguancun
Beijing

Phone: +86 10 58813007
EMail: yaojk@cnnic.cn

Kazunori Fujiwara
Japan Registry Services Co., Ltd.
Chiyoda First Bldg. East 13F, 3-8-1 Nishi-Kanda
Tokyo

Phone: +81 3 5215 8451
EMail: fujiwara@jprs.co.jp

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 8, 2011

A. Melnikov, Ed.
Isode Ltd

March 7, 2011

Internationalized Email Addresses in X.509 certificates
draft-ietf-pkix-eai-addresses-00

Abstract

This document defines a new name form for inclusion in the otherName field of an X.509 Subject Alternative Name extension that allows a certificate subject to be associated with an Internationalized Email Address.

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 September 8, 2011.

Table of Contents

1. Introduction

[RFC5280] defines rfc822Name subjectAltName choice for representing [RFC5322] email addresses. This form is restricted to a subset of US-ASCII characters and thus can't be used to represent Internationalized Email addresses [I-D.ietf-eai-rfc5336bis].

2. Conventions Used in This Document

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].

The formal syntax use the Augmented Backus-Naur Form (ABNF) [RFC5234] notation.

3. Name Definitions

This section defines the eaiName name as a form of otherName from the GeneralName structure in SubjectAltName defined in [RFC5280].

```
id-on-eaiAddr OBJECT IDENTIFIER ::= { id-on XXX }

eaiName ::= UTF8String (SIZE (1..MAX))
```

When the subjectAltName extension contains an Internationalized Email address, the address MUST be stored in the eaiName name form of otherName. The format of an eaiName is a <eaiMailbox> as defined below. A eaiMailbox has the form "Local-part@Domain". Note that a eaiMailbox has no phrase (such as a common name) before it, has no comment (text surrounded in parentheses) after it, and is not surrounded by "<" and ">".

```
eaiMailbox = uLocal-part "@" uDomain
```

uLocal-part and uDomain are defined in [I-D.ietf-eai-rfc5336bis].

4. Matching of Internationalized Email Addresses in X.509 certificates

The <uLocal-part> part of an Internationalized email address is in UTF-8 and need to be compared octet for octet.

The <uDomain> can contain either IDN domain or an ASCII Compatible Encoding (ACE) format. When comparing two <uDomain>s both MUST be converted to the ACE form as described in section 7.2 of [RFC5280].

5. IANA Considerations

[[anchor6: Just need a new OID.]]

6. Security Considerations

[[anchor7: TBD]]

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.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[I-D.ietf-eai-rfc5336bis] Yao, J. and W. Mao, "SMTP extension for internationalized email address", draft-ietf-eai-rfc5336bis-08 (work in progress), March 2011.

7.2. Informative References

[RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, October 2008.

Appendix A. Acknowledgements

Thank you to Magnus Nystrom for motivating this document.

Author's Address

Alexey Melnikov (editor)
Isode Ltd
5 Castle Business Village
36 Station Road
Hampton, Middlesex TW12 2BX
UK

Email: Alexey.Melnikov@isode.com