

Internet Engineering Task Force
Internet-Draft
Obsoletes: RFC5738 (if approved)
Intended status: Standards Track
Expires: January 17, 2013

P. Resnick, Ed.
Qualcomm Incorporated
C. Newman, Ed.
Oracle
S. Shen, Ed.
CNNIC
July 16, 2012

IMAP Support for UTF-8
draft-ietf-eai-5738bis-06

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. This specification replaces RFC 5738.

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 January 17, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Conventions Used in this Document	3
3. UTF8=ACCEPT IMAP Capability	3
3.1. UTF-8 in IMAP Quoted Strings	4
4. IMAP UTF8 Append Data Extension	5
5. LOGIN Command and UTF-8	5
6. UTF8=ONLY Capability	6
7. Dealing With Legacy Clients	6
8. Issues with UTF-8 Header Mailstore	7
9. IANA Considerations	7
10. Security Considerations	7
11. References	8
11.1. Normative References	8
11.2. Informative References	9
Appendix A. Appendix A. Design Rationale	9
Appendix B. Appendix B. Acknowledgments	10

1. Introduction

This specification extends IMAP4rev1 [RFC3501] to permit UTF-8 [RFC3629] in headers as described in "Internationalized Email Headers" [RFC6532]. It also adds a mechanism to support mailbox names using the UTF-8 charset. This specification creates two new IMAP capabilities to allow servers to advertise these new extensions.

Most of this specification assumes that the IMAP server will be operating in a fully internationalized environment, i.e., one in which all clients accessing the server will be able to accept non-ASCII message header fields and other information as specified in Section 3. At least during a transition period, that assumption will not be realistic for many environments; the issues involved are discussed in Section 7 below.

This specification replaces an earlier, experimental, approach to the same problem [RFC5738].

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. 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 the ability to open mailboxes containing internationalized messages with 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 in quoted-strings. The "ENABLE UTF8=ACCEPT" command MUST only be used in the authenticated state. (Note that the "UTF8=ONLY" capability described in Section 6 imply the "UTF8=ACCEPT" capability. See

additional information in these sections.)

3.1. UTF-8 in IMAP 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 UTF-8 in quoted-strings with the following syntax:

```
quoted          =/ DQUOTE *uQUOTED-CHAR DQUOTE
                  ; QUOTED-CHAR is not modified, as it will affect
                  ; other RFC 3501 ABNF non terminal.

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 extended quoting mechanism is used by the client, 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 UTF-8 in quoted strings 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 extended 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. 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 UTF-8 in quoted strings 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).

4. IMAP UTF8 Append Data Extension

If the "UTF8=ACCEPT" 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 ")"
literal8       = <Defined in RFC 4466>
append-data    =/ utf8-literal
cat-part       =/ utf8-literal
```

IMAP servers that advertise support for "UTF8=ACCEPT" or "UTF8=ONLY" MUST reject an APPEND command that includes any 8-bit in the message headers with a "NO" response, when IMAP clients do not issue "ENABLE UTF8=ACCEPT" or "ENABLE UTF8=ONLY".

Note that the "UTF8=ONLY" capability described in Section 6 implies the "UTF8=ACCEPT" capability. See additional information in that section.

5. LOGIN Command and UTF-8

This specification doesn't extend the IMAP LOGIN command [RFC3501] to support UTF-8 usernames and passwords. Whenever a client needs to use UTF-8 username/passwords, it MUST use the IMAP AUTHENTICATE command which is already capable of passing UTF-8 user names and credentials.

Although this makes it syntactically legal to have a UTF-8 user name or password, there is no guarantee the user provisioning system used by the IMAP server will allow such identities. This is an implementation decision and MAY depend on what identity system the

IMAP server is configured to use.

6. 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). 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. UTF8=ACCEPT and UTF8=ONLY SHOULD be mutually exclusive. An IMAP server can advertise one of them, but never both.

7. Dealing With Legacy Clients

In most situations, it will be difficult or impossible for the implementer or operator of an IMAP (or POP) server to know whether all of the clients that might access it, or the associated mail store more generally, will be able to support the facilities defined in this document. In almost all cases, servers who conform to this specification will have to be prepared to deal with clients that do not enable the relevant capabilities. Unfortunately, there is no completely satisfactory way to do so other than for systems that wish to receive email that requires SMTPUTF8 capabilities to be sure that all components of those systems -- including IMAP and other clients selected by users -- are upgraded appropriately.

Choices available to the server when a message that requires SMTPUTF8 is encountered and the client doesn't enable UTF-8 capability include hiding the problematic message(s) as outlined elsewhere in this specification, creating in band or out of band notifications or error messages, or somehow trying to create a variation on the message with the intention of providing useful information to that client about what has occurred. Such variant messages cannot be actual substitutes for the original message: it will rarely be possible to reply to (either at all or without loss of information), new header fields or specialized constructs for server-client communication may go beyond the requirements of, e.g., RFC 5322 and may consequently confuse some legacy mail user agents (including IMAP clients) or otherwise not provide the expected information to users. There are also tradeoffs in constructing variants of the original message between accepting complexity and additional computation costs in order to try to preserve as much information as possible (for example, in [popimap-downgrade]) and trying to minimize those costs while still providing useful information (for example, in

[I-D.ietf-eai-simplifieddowngrade]]).

Because such messages are really variations on the original ones, not really "downgraded" (ones although that terminology is often used for convenience), they inevitably have relationships to the original ones that the IMAP specification [RFC3501] did not anticipate. In particular, digital signatures computed over the original message will often not be applicable to the variant version and servers that may be accessed by the same user with different clients or methods (e.g., POP or webmail systems in addition to IMAP or IMAP clients with different capabilities) will need to exert extreme care to be sure that UIDVALIDITY behaves as the user would expect. Those issues may be especially sensitive if the server caches the variant message or computes and stores it when the message arrives with the intent of making either form available depending on client capabilities.

The best (or "least bad") approach for any given environment will depend on local conditions, local assumptions about user behavior, the degree of control the server operator has over client usage and upgrading, the options that are actually available, and so on. It is impossible, at least at the time, to give good advice that will apply to all situations, or even particular profiles of situations, other than "upgrade legacy clients as soon as possible".

8. 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].

9. IANA Considerations

This document adds two new capabilities ("UTF8=ACCEPT" and "UTF8=ONLY") to the IMAP4rev1 Capabilities registry [RFC3501]. Three other IMAP capabilities that were described in the experimental predecessor to this document (UTF8=ALL, UTF8=APPEND, UTF8=USER) are to be marked OBSOLETE in the registry.

10. 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.

Special considerations, some of them with security implications, occur if a server that conforms to this specification is accessed by a client that does not and in some more complex situations in which a given message is accessed by multiple clients that might use different protocols and/or support different capabilities. Those issues are discussed in Section 7 above.

11. References

11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5198] Klensin, J. and M. Padlipsky, "Unicode Format for Network Interchange", RFC 5198, March 2008.
- [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.
- [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.
- [RFC6532] Yang, A., Steele, S., and N. Freed, "Internationalized Email Headers", RFC 6532, February 2012.
- [RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, October 2008.

11.2. Informative References

- [RFC2088] Myers, J., "IMAP4 non-synchronizing literals", RFC 2088, January 1997.
- [RFC5738] Resnick, P. and C. Newman, "IMAP Support for UTF-8", RFC 5738, March 2010.
- [I-D.ietf-eai-simplifiedowngrade] Gulbrandsen, A., "EAI: Simplified POP/IMAP downgrading", draft-ietf-eai-simplifiedowngrade-05 (work in progress), June 2012.
- [popimap-downgrade] Fujiwara, K., "Post-delivery Message Downgrading for Internationalized Email Messages", draft-ietf-eai-popimap-downgrade-06 (work in progress), July 2012.

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.

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.

Appendix B. Appendix B. 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 Hurtt, 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

EAI
Internet-Draft
Intended status: Informational
Expires: December 31, 2012

J. Levine
Taughannock Networks
R. Gellens
Qualcomm Incorporated
July 2012

Mailing Lists and non-ASCII Addresses
draft-ietf-eai-mailinglistbis-05

Abstract

This document describes considerations for mailing lists with the introduction of non-ASCII UTF-8 email addresses. It outlines some possible scenarios for handling lists with mixtures of non-ASCII and traditional addresses, but does not specify protocol changes or offer implementation or deployment advice.

NOTE TO REVIEWERS: Missing or odd-looking references between sections are due to bugs in xml2rfc. The XML is OK, and the HTML output looks reasonable.

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 December 31, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components

extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Mailing list header additions and modifications	3
1.2. Non-ASCII email addresses	3
2. Scenarios Involving Mailing Lists	4
2.1. Fully SMTPUTF8 lists	4
2.2. Mixed SMTPUTF8 and ASCII lists	5
2.3. SMTP issues	5
3. List headers	6
3.1. SMTPUTF8 list headers	6
3.2. Downgrading list headers	7
3.3. Subscribers' addresses in downgraded headers	8
4. Security considerations	8
5. IANA considerations	8
6. References	8
6.1. Normative References	8
6.2. Informative References	8
Appendix A. Change Log	9
Appendix A.1. Change from -04 to -05	9
Appendix A.2. Change from -03 to -04	9
Appendix A.3. Change from -02 to -03	9
Appendix A.4. Changes up to -02	9
Appendix A.5. Changes up to -00	9
Authors' Addresses	9

1. Introduction

This document describes considerations for mailing lists with the introduction of non-ASCII UTF-8 email addresses. The usage of such addresses is described in [RFC6530].

Mailing lists are an important part of email usage and collaborative communications. The introduction of internationalized email addresses affects mailing lists in three main areas: (1) transport (receiving and sending messages); (2) message headers of received and retransmitted messages; and (3) mailing list operational policies.

A mailing list is a mechanism that distributes a message to multiple recipients when the originator sends it to a single address. An agent, usually software rather than a person, at that single address receives the message and then causes the message to be redistributed to a list of recipients. This agent usually sets the envelope return address (henceforth called the bounce address) of the redistributed message to a different address from that of the original message. Using a different bounce address directs error and other automatically generated messages to an error handling address associated with the mailing list. This sends error and other automatic messages to the list agent, which can often do something useful with them, rather than to the original sender, who typically doesn't control the list and hence can't do anything about them.

In most cases, the mailing list agent redistributes a received message to its subscribers as a new message, that is, conceptually it uses message submission [RFC6409] (as did the sender of the original message). The exception, where the mailing list is not managed by a separate agent that receives and redistributes messages in separate transactions, but is implemented by an expansion step within an SMTP transaction where one local address expands to multiple local or non-local addresses, is not addressed by this document.

1.1. Mailing list header additions and modifications

Some list agents alter message header fields, while others do not. A number of standardized list-related header fields have been defined, and many lists add one or more of these headers. Separate from these standardized list-specific header fields, and despite a history of interoperability problems from doing so, some lists alter or add header fields in an attempt to control where replies are sent. Such lists typically add or replace the "Reply-To" field and some add or replace the "Sender" field. Some lists alter or replace other fields, including "From".

Among these list-specific header fields are those specified in RFCs 2369 [RFC2369] and 2919 [RFC2919]. For more information, see Section 3.

1.2. Non-ASCII email addresses

While the mail transport protocol is the same for regular email recipients and mailing list recipients, list agents have special considerations with non-ASCII email addresses because they retransmit messages composed by other agents to potentially many recipients.

There are considerations for non-ASCII email addresses in the envelope as well as in header fields of redistributed messages. In particular, a message with non-ASCII addresses in the headers or envelope cannot be sent to non-SMTPUTF8 recipients.

With mailing lists, there are two different types of considerations: first, the purely technical ones involving message handling, error cases, and the like, and second, those that arise from the fact that humans use mailing lists to communicate. As an example of the first, list agents might choose to reject all messages from non-ASCII addresses if they are unprepared to handle SMTPUTF8 mail. As an example of the second, a user who sends a message to a list often is unaware of the list membership. In particular, the user often doesn't know if the members are SMTPUTF8 mail users or not, and often neither the original sender nor the recipients personally know each other. As a consequence of this, remedies that may be readily available for one-to-one communication might not be appropriate when dealing with mailing lists. For example, if a user sends a message which is undeliverable, normally the telephone, instant messaging, or other forms of communication are available to obtain a working address. With mailing lists, the users may not have any recourse. Of course, with mailing lists, the original sender usually does not know which list members successfully received a message, or if it was undeliverable to some.

Conceptually, a mailing list's internationalization can be divided into three capabilities: First, does the list have a non-ASCII submission address? Second, does the list agent accept subscriptions for addresses containing non-ASCII characters? And third, does the list agent accept messages that require SMTPUTF8 capabilities?

If a list has subscribers with ASCII addresses, those subscribers might or might not be able to accept SMTPUTF8 messages.

2. Scenarios Involving Mailing Lists

Generally (and exclusively within the scope of this document), an original message is sent to a mailing list as a completely separate and independent transaction from the list agent sending the retransmitted message to one or more list recipients. In both cases, the message might be addressed only to the list address, or might have recipients in addition to the list. Furthermore, the list agent might choose to send the retransmitted message to each list recipient in a separate message submission transaction, or might choose to include multiple recipients per transaction. Often, list agents are constructed to work in cooperation with, rather than include the functionality of, a message submission server, and hence the list transmits to a single submission server one copy of the retransmitted message. The submission server then decides which recipients to include in which transaction.

2.1. Fully SMTPUTF8 lists

Some lists may wish to be fully SMTPUTF8. That is, all subscribers are expected to be able to receive SMTPUTF8 mail. For list hygiene reasons, such a list would probably want to prevent subscriptions from addresses that are unable to receive SMTPUTF8 mail. If a putative subscriber has a non-ASCII address, it must be able to receive SMTPUTF8 mail, but there is no way to tell whether a subscriber with an ASCII address can receive SMTPUTF8 mail short of sending an SMTPUTF8 probe or confirmation message and somehow finding out whether it was delivered, e.g., if the user clicked a link in the confirmation message.

2.2. Mixed SMTPUTF8 and ASCII lists

Other lists may wish to handle a mixture of SMTPUTF8 and ASCII subscribers, either as a transitional measure as subscribers upgrade to SMTPUTF8-capable mail software, or as an ongoing feature. While it is not possible in general to downgrade SMTPUTF8 mail to ASCII mail, list software might divide the recipients into two sets, SMTPUTF8 and ASCII recipients, and create a downgraded version of SMTPUTF8 list messages to send to ASCII recipients. See Section 3.2 and Section 3.3.

To determine which set an address belongs in, list software might make the conservative assumption that ASCII addresses get ASCII messages, it might try to probe the address with an SMTPUTF8 test message, or it might let the subscriber set the message format manually, similar to the way that some lists now let subscribers choose between plain text and HTML mail, or individual messages and a daily digest.

To determine whether a message needs to be downgraded for ASCII recipients, list software might assume that any message received via an SMTPUTF8 SMTP session is an SMTPUTF8 message, or might examine the headers and body of the message to see whether it needs SMTPUTF8 treatment. Depending on the interface between the list software and the MTA and MDA that handle incoming messages, it may not be able to tell the type of session for incoming messages.

2.3. SMTP issues

Mailing list software usually changes the envelope addresses on each message. The bounce address is set to an address that will return bounces to the list agent, and the recipient addresses are set to the subscribers of the list. For some lists, all messages to a list get the same bounce address. For others, list software may create a

bounce address per recipient, or a unique bounce address per message per recipient, bounce management techniques known as Variable Envelope Return Path or VERP [VERP].

The bounce address for a list typically includes the name of the list, so a list with a non-ASCII name will have a non-ASCII bounce address. Given the unknown paths that bounce messages might take, list software might instead use an ASCII bounce address to make it more likely that bounces can be delivered back to the list agent. Similarly, a VERP address for each subscriber typically embeds a version of the subscriber's address so the VERP bounce address for a non-ASCII subscriber address will be a non-ASCII address. For the same reason, the list software might use ASCII bounce addresses that encode the recipient's identity in some other way.

3. List headers

List agents typically adds list-specific headers to each message before resending the message to list recipients.

3.1. SMTPUTF8 list headers

The list headers in RFCs 2369 [RFC2369] and 2919 [RFC2919] were all specified before SMTPUTF8 mail existed and their definitions do not address where non-ASCII characters might appear. These include, for example:

```
List-Id: List Header Mailing List
      <list-header.example.com>
List-Help:
      <mailto:list@example.com?subject=help>
List-Unsubscribe:
      <mailto:list@example.com?subject=unsubscribe>
List-Subscribe:
      <mailto:list@example.com?subject=subscribe>
List-Post:
      <mailto:list@example.com>
List-Owner:
      <mailto:listmom@example.com> (Contact Person for Help)

List-Archive: <mailto:archive@example.com?subject=index%20list>
```

As described in [RFC2369], "The contents of the list header fields mostly consist of angle-bracket ('<', '>') enclosed URLs, with internal whitespace being ignored." [RFC2919] specifies that "The list identifier will, in most cases, appear like a host name in a domain of the list owner." Since these headers were defined in the context of ASCII mail, these headers permit only ASCII text including in the URLs.

The most commonly-used URI schemes in List-* headers tend to be http and mailto [RFC6068], although they sometimes include https and ftp, and in principle can contain any valid URI.

Even if a scheme permits an internationalized form, it should use a pure ASCII form of the URI described in [RFC3986]. Future work may extend these header fields or define replacements to directly support unencoded non-ASCII outside the ASCII repertoire in these and other header fields, but in the absence of such extension or replacement, non-ASCII characters can only be included by encoding them as ASCII.

The encoding technique specified in [RFC3986] is to use a pair of hex digits preceded by a percent sign, but percent signs have been used informally in mail addresses to do source routing. Although few mail systems still permit source routing, a lot of mail software still forbids or escapes characters formerly used for source routing, which can lead to unfortunate interactions with percent-encoded URIs or any URI that includes one of those characters. If a program interpreting a mailto: URI knew that the MUA in use were able to handle non-ASCII data, the program could pass the URI in unencoded non-ASCII, avoiding problems with misinterpreted percent signs, but at this point there is no standard or even informal way for MUAs to signal SMTPUTF8 capabilities. Also, note that whether internationalized domain names should be percent-encoded or puny-coded depends on the context in which they occur.

The List-ID header field uniquely identifies a list. The intent is that the value of this header remain constant, even if the machine or system used to operate or host the list changes. This header field is often used in various filters and tests, such as client-side filters, Sieve filters [RFC5228], and so forth. If the definition of a List-ID header field were to be extended to allow non-ASCII text, filters and tests might not properly compare encoded and unencoded versions of a non-ASCII value. In addition to these comparison considerations, it is generally desirable that this header field contain something meaningful that users can type in. However, ASCII encodings of non-ASCII characters are unlikely to be meaningful to users or easy for them to accurately type.

3.2. Downgrading list headers

If list software prepares a downgraded version of an SMTPUTF8 message, all the List-* headers must be downgraded. In particular, if a List-* header contains a non-ASCII mailto (even encoded in ASCII), it may be advisable to edit the header to remove the non-ASCII address, or replace it with an equivalent ASCII address if one is known to the list software. Otherwise, a client might run into trouble if the decoded mailto results in a non-ASCII address. If a header that contains a mailto URL is downgraded by percent encoding, some mail software may misinterpret the percent signs as attempted source routing.

When downgrading list headers, it may not be possible to produce a downgraded version that is satisfactorily equivalent to the original header. In particular, if a non-ASCII List-ID is downgraded to an ASCII version, software and humans at recipient systems will typically not be able to tell that both refer to the same list.

If lists permit mail with multiple MIME parts, some MIME headers in SMTPUTF8 messages may include non-ASCII characters in file names and other descriptive text strings. Downgrading these strings may lose the sense of the names, break references from other MIME parts (such as HTML IMG references to embedded images) and otherwise damage the mail.

3.3. Subscribers' addresses in downgraded headers

List software typically leaves the original submitter's address in the From: line, both so that recipients can tell who wrote the message, and so that they have a choice of responding to the list or directly to the submitter. If a submitter has a non-ASCII address, there is no way to downgrade the From: header and preserve the address so that ASCII recipients can respond to it, since non-SMTPUTF8 mail systems can't send mail to non-ASCII addresses.

Possible work arounds (none implemented that we know of) might include allowing subscribers with non-ASCII addresses to register an alternate ASCII address with the list software, having the list software itself create ASCII forwarding addresses, or just putting the list's address in the From: line and losing the ability to respond directly to the submitter.

4. Security considerations

None beyond what mailing list agents do now.

5. IANA considerations

NOTE TO RFC EDITOR: This section may be removed upon publication of this document as an RFC.

This document makes no requests to IANA.

6. References

6.1. Normative References

- [RFC3986] Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.
- [RFC6068] Duerst, M., Masinter, L. and J. Zawinski, "The 'mailto' URI Scheme", RFC 6068, October 2010.
- [RFC6409] Gellens, R. and J. Klensin, "Message Submission for Mail", STD 72, RFC 6409, November 2011.
- [RFC6530] Klensin, J. and Y. Ko, "Overview and Framework for Internationalized Email", RFC 6530, February 2012.

6.2. Informative References

- [RFC2369] Neufeld, G. and J.D. Baer, "The Use of URLs as Meta-Syntax for Core Mail List Commands and their Transport through Message Header Fields", RFC 2369, July 1998.
- [RFC2919] Chandhok, R. and G. Wenger, "List-Id: A Structured Field and Namespace for the Identification of Mailing Lists", RFC 2919, March 2001.
- [RFC5228] Guenther, P. and T. Showalter, "Sieve: An Email Filtering Language", RFC 5228, January 2008.
- [VERP] "Variable Envelope Return Path", .

Appendix A. Change Log

NOTE TO RFC EDITOR: This section may be removed upon publication of this document as an RFC.

Appendix A.1. Change from -04 to -05

 Add place holder IANA considerations

Appendix A.2. Change from -03 to -04

 Update references

Appendix A.3. Change from -02 to -03

 Distinguish lists from agents.

 Change refs to EAI to non-ASCII addresses or SMTPUTF8 mail capabilities.

 Reference for VERP

 Clarify discussions of IRIs.

 Capitalize Mailto and http Consistently.

Appendix A.4. Changes up to -02

 Various editorial changes.

 Refer to RFC 6068.

Appendix A.5. Changes up to -00

 Rewrite completely.

Authors' Addresses

John Levine
Taughannock Networks
PO Box 727
Trumansburg, NY 14886

Phone: +1 831 480 2300
Email: standards@taugh.com
URI: <http://jl.ly>

Randall Gellens
Qualcomm Incorporated
5775 Morehouse Drive
San Diego, CA 92121

Email: rg+ietf@qualcomm.com

Email Address Internationalization
(EAI)
Internet-Draft
Intended status: Standards Track
Expires: January 10, 2013

K. Fujiwara
JPRS
July 9, 2012

Post-delivery Message Downgrading for Internationalized Email Messages
draft-ietf-eai-popimap-downgrade-06.txt

Abstract

The Email Address Internationalization (SMTPUTF8) extension 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 send 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 treat 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 January 10, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal

Provisions Relating to IETF Documents
(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	4
1.1. Problem statement	4
1.2. Possible solutions	4
1.3. Approach taken in this specification	4
2. Terminology	5
3. New Header Fields Definition	6
4. Email Header Fields Downgrading	6
4.1. Downgrading Method for Each ABNF Element	7
4.1.1. <UNSTRUCTURED> Downgrading	7
4.1.2. <WORD> Downgrading	7
4.1.3. <COMMENT> Downgrading	7
4.1.4. <MIME-VALUE> Downgrading	7
4.1.5. <DISPLAY-NAME> Downgrading	7
4.1.6. <GROUP> Downgrading	7
4.1.7. <MAILBOX> Downgrading	8
4.1.8. ENCAPSULATION Downgrading	8
4.1.9. <TYPED-ADDRESS> Downgrading	8
4.2. Downgrading Method for Each Header Field	9
4.2.1. Address Header Fields That Contain <address>s	9
4.2.2. Address Header Fields with Typed Addresses	9
4.2.3. Downgrading Non-ASCII in Comments	10
4.2.4. Message-ID Header Fields	10
4.2.5. Received Header Field	10
4.2.6. MIME Content Header Fields	10
4.2.7. Non-ASCII in <unstructured>	11
4.2.8. Non-ASCII in <phrase>	11
4.2.9. Other Header Fields	11
5. MIME Body-Part Header Field Downgrading	11
6. Security Considerations	12
7. Implementation Notes	13
7.1. RFC 2047 Encoding	13
8. IANA Considerations	13
9. Acknowledgements	14
10. Change History	14
10.1. Version 00	15
10.2. Version 01	15
10.3. Version 02	15

10.4. Version 03	15
10.5. Version 04	15
10.6. Version 05	16
10.7. Version 06	16
11. References	16
11.1. Normative References	16
11.2. Informative References	17
Appendix A. Examples	18
A.1. Downgrading Example	18

1. Introduction

1.1. Problem statement

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

If a header field contains non-ASCII strings, POP/IMAP servers cannot send Internationalized Email Headers to legacy clients 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.
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.

This message downgrading mechanism converts mail header fields to an all-ASCII representation. The POP/IMAP servers can use the downgrading mechanism and send the Internationalized Email message as

a traditional form. Receivers can know they received some internationalized messages or some unknown/broken messages.

[RFC6532] allows UTF-8 characters to be used in mail header fields and MIME 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:

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

In Section 3 of this document, header fields starting with "Downgraded-" are introduced. They preserve the information that appeared in the original header fields.

Email header field downgrading is described in Section 4. It generates ASCII-only header fields.

The definition of MIME header fields in Internationalized Email Messages is described in [RFC6532]. MIME header field downgrading is described in Section 5. 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 extension 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 "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. New Header Fields Definition

New header fields are defined to preserve information that appeared in non-ASCII strings in header fields of the incoming message. The values of the new fields holds the original header field value in encoded form. The revised header field syntax is specified as follows:

```
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
```

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.

4. Email Header Fields Downgrading

This section defines the conversion method to ASCII for each header field that may contain non-ASCII strings.

[RFC6532] expands "Received:" header fields; [RFC5322] describes ABNF elements <mailbox>, <word>, <comment>, <unstructured>; [RFC2045] describes ABNF element <value>.

4.1. 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.

4.1.1. <UNSTRUCTURED> Downgrading

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

4.1.2. <WORD> Downgrading

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

4.1.3. <COMMENT> Downgrading

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

4.1.4. <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.

4.1.5. <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.

4.1.6. <GROUP> Downgrading

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

If the header field has any <group> elements that contain <mailbox> elements, and those <mailbox> elements in turn contain non-ASCII addresses, rewrite each <group> element as

display-name ENCODED_WORD " :;"

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

4.1.7. <MAILBOX> Downgrading

The <mailbox> elements have no equivalent format for non-ASCII addresses. If the header field has any <mailbox> elements that contain non-ASCII strings in their <addr-spec> element, rewrite each <addr-spec> element to ASCII-only format. 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].

4.1.8. ENCAPSULATION Downgrading

Encapsulate the header field in a "Downgraded-" header field as described in Section 3 as a last resort.

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.

4.1.9. <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 downgrading on the entire header field.

4.2. 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.

4.2.1. 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 <group> elements that contain non-ASCII addresses, perform <COMMENT> downgrading, <DISPLAY-NAME> downgrading, and <GROUP> downgrading.

If the header field contains <mailbox> elements that contain non-ASCII addresses, perform <COMMENT> downgrading, <DISPLAY-NAME> downgrading, and <MAILBOX> downgrading.

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.

4.2.2. Address Header Fields with Typed Addresses

Original-Recipient:

Final-Recipient:

If the header field contains non-ASCII strings, perform <TYPED-ADDRESS> downgrading.

4.2.3. 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.

4.2.4. Message-ID Header Fields

Message-ID:

Resent-Message-ID:

In-Reply-To:

References:

Perform ENCAPSULATION Downgrading.

4.2.5. Received Header Field

Received:

If the FOR clause contains a non-ASCII address, remove the FOR clause from the header field. Comments may contain non-ASCII strings, Perform <COMMENT> downgrading. Other parts should not contain non-ASCII strings.

4.2.6. MIME Content Header Fields

Content-Type:
Content-Disposition:

Perform <MIME-VALUE> downgrading and <COMMENT> downgrading.

4.2.7. Non-ASCII in <unstructured>

Subject:
Comments:
Content-Description:

Perform <UNSTRUCTURED> downgrading.

4.2.8. Non-ASCII in <phrase>

Keywords:

Perform <WORD> downgrading.

4.2.9. 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.

5. 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 4.

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.

6. 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 which traditional MUAs do not care.

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). Receivers may know they need to update their MUAs, or they don't care.

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 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.

7. Implementation Notes

7.1. 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.

8. IANA Considerations

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

[RFC5504] registered many "Downgraded-" header fields and requested that 'IANA will refuse registration of all field names that start with "Downgraded-", to avoid possible conflict with the procedure for unknown header fields' preservation described in Section 3.3 of [RFC5504].' However [RFC6530] obsoleted [RFC5504] and this document does not use all "Downgraded-" header fields registered by [RFC5504].

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)

Header field name: Downgraded-In-Reply-To
Applicable protocol: mail
Status: standard
Author/change controller: IETF
Specification document(s): This document (Section 3)

Header field name: Downgraded-References
Applicable protocol: mail
Status: standard
Author/change controller: IETF
Specification document(s): This document (Section 3)

Header field name: Downgraded-Original-Recipient
Applicable protocol: mail
Status: standard
Author/change controller: IETF
Specification document(s): This document (Section 3)

Header field name: Downgraded-Final-Recipient
Applicable protocol: mail
Status: standard
Author/change controller: IETF
Specification document(s): This document (Section 3)

9. 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.

10. 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.

10.1. Version 00

- o Initial version
- o Imported header field downgrading from RFC 5504

10.2. Version 01

- o same as Version 00

10.3. Version 02

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

10.4. Version 03

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

10.5. Version 04

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

10.6. Version 05

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

10.7. Version 06

- o Removed "Updating RFC 5322"
- o Added reference to draft-leiba-5322upd-from-group

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 (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. |

- [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.
- [RFC6530] Klensin, J. and Y. Ko, "Overview and Framework for Internationalized Email", RFC 6530, 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:' Header Field", draft-leiba-5322upd-from-group-01 (work in progress), July 2012.

11.2. 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:

- o The sender address is a non-ASCII address, "NON-ASCII-LOCAL@example.com". Its display-name is "DISPLAY-LOCAL".
- o 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".
- o The "Cc:" header field contains a non-ASCII address, "NON-ASCII-REMOTE3@example.org". Its display-name is "DISPLAY-REMOTE3".
- o Four display names contain non-ASCII characters.
- o The Subject header field is "NON-ASCII-SUBJECT", which contains non-ASCII strings.
- o The "Message-Id:" header field contains "NON-ASCII-MESSAGE_ID", which contains non-ASCII strings.
- o 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: DATE
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: DATE
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

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@wide.ad.jp, fujiwara@jprs.co.jp

Network Working Group
Internet-Draft
Obsoletes: 5721 (if approved)
Intended status: Standards Track
Expires: January 17, 2013

R. Gellens
QUALCOMM Incorporated
C. Newman
Oracle
J. Yao
CNNIC
K. Fujiwara
JPRS
July 16, 2012

POP3 Support for UTF-8
draft-ietf-eai-rfc5721bis-06.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 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 January 17, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	3
1.1. Conventions Used in This Document	3
2. LANG Capability	4
3. UTF8 Capability	6
3.1. The UTF8 Command	7
3.2. USER Argument to UTF8 Capability	8
4. Native UTF-8 Maildrops	9
5. UTF8 Response Code	9
6. IANA Considerations	9
7. Security Considerations	10
8. Change History	10
8.1. draft-ietf-eai-rfc5721bis: Version 00	10
8.2. draft-ietf-eai-rfc5721bis: Version 01	10
8.3. draft-ietf-eai-rfc5721bis: Version 02	10
8.4. draft-ietf-eai-rfc5721bis: Version 03	10
8.5. draft-ietf-eai-rfc5721bis: Version 04	11
8.6. draft-ietf-eai-rfc5721bis: Version 05	11
9. References	11
9.1. Normative References	11
9.2. Informative References	12
Appendix A. Design Rationale	12
Appendix B. Acknowledgments	13

1. Introduction

This document forms part of the Email Address Internationalization (EAI) protocols described in the EAI Framework document [RFC6530]. As part of the overall EAI work, email messages could be transmitted and delivered containing un-encoded UTF-8 characters in the header and/or body, and maildrops 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, and bodies (e.g., transferred using 8-bit Content Transfer Encoding) as described in "Internationalized Email Headers" [RFC6532]. It also adds a mechanism to support login names and passwords containing UTF-8 characters, and a mechanism to support UTF-8 characters in protocol level response strings as well as the ability to negotiate a language for such response strings.

This specification also adds a new response code to indicate that a message could not be returned because it requires UTF-8 mode and the server is unwilling to create and deliver variant form of the message discussed in Section 7 of [I-D.ietf-eai-5738bis].

This specification replaces an earlier, experimental, approach to the same problem RFC 5721 [RFC5721]. Section 6 of [RFC6530] describes the changes in approach between RFC 5721 [RFC5721] and this specification.

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].

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; Otherwise, some examples for the "LANG" command may appear incorrectly 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:

AUTHORIZATION, 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 uses when sending human-readable text.

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.

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. There are no specific listing order of languages, which may depend on configuration or implementation.

Examples:

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

```
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: .
```

< A request for a language listing might fail. >

C: LANG

S: -ERR Server is unable to list languages

< Once the client selects 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. In UTF-8 mode, both servers and clients can send and accept UTF-8 characters.

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 [RFC6532] 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. If a client requests a UTF-8 message when not in UTF-8 mode, the server MUST either create the message content variant (discussed in Section 7 of [I-D.ietf-eai-5738bis]) it sends to the client to comply with unextended POP and Internet Mail Format without UTF-8 mode support, or fail the request with a -ERR response containing the UTF-8 response code (see section 5). The UTF8 command MAY fail.

Note that even in UTF-8 mode, MIME binary content-transfer-encoding as defined in MIME Section 6.2 [RFC2045] 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 were.

Normal operation for UTF-8 maildrops will be for both servers and clients to support the extension discussed in this specification. Upgrading of both clients and servers is the only fully satisfactory way to support the capabilities offered by the "UTF8" extension and SMTPUTF8 mail more generally. Servers must, however, anticipate the possibility of a client attempting to access a message that requires

this extension without having issued the "UTF8" command. There are no completely satisfactory responses for that case other than upgrading the client to support this specification. One solution, unsatisfactory because the user may be confused by being able to access the message through some means and not others, is that a server MAY choose to reject the command to retrieve the message as discussed in Section 5. Other alternatives, including the possibility of creating and delivering variant form of the message, are discussed in Section 7 of [I-D.ietf-eai-5738bis].

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] or one of its standards-track successors 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] or one of its standards-track successors 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 [RFC3454] (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 [RFC3454] (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. When the server is not in UTF-8 mode and the message requires that mode, requests to download the message MAY be rejected (as specified in the next section) or the various other alternatives outlined in Section 3.1 above and in Section 7 of the IMAP UTF-8 specification [draft-ietf-eai-5738bis], including creation and delivery of variations on the original message, MAY be considered.

5. UTF8 Response Code

Per "POP3 Extension Mechanism" [RFC2449], this document adds a new response code: UTF8, described below.

Complete response code:
UTF8

Valid for responses:
-ERR

Valid for commands:
LIST, TOP, RETR

Response code meaning and expected client behavior:

The UTF8 response code indicates that a failure is due to a request when not in UTF-8 mode for message content containing UTF-8 characters.

The client MAY reissue the command after entering UTF-8 mode.

6. IANA Considerations

Section 2 and 3 of this specification update two capabilities ("UTF8" and "LANG") to the POP3 capability registry [RFC2449].

Section 5 of this specification also adds one new response code ("UTF8") to the POP3 response codes registry [RFC2449].

7. 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 are strongly advised to 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 protect the integrity of 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).

8. Change History

8.1. draft-ietf-eai-rfc5721bis: Version 00

following the new charter

8.2. draft-ietf-eai-rfc5721bis: Version 01

refine the texts

8.3. draft-ietf-eai-rfc5721bis: Version 02

update the texts based on Joseph's comments

8.4. draft-ietf-eai-rfc5721bis: Version 03

improve the texts

text instructing servers to either downconvert or reject

new UTF-8 response code for use

8.5. draft-ietf-eai-rfc5721bis: Version 04

improve the texts

8.6. draft-ietf-eai-rfc5721bis: Version 05

updated according to jabber interim meeting result

updated according to john and apparea's review comments

9. References

9.1. Normative References

- [I-D.ietf-eai-5738bis] Resnick, P., Newman, C., and S. Shen, "IMAP Support for UTF-8", draft-ietf-eai-5738bis-03 (work in progress), December 2011.
- [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.
- [RFC2449] Gellens, R., Newman, C., and L. Lundblade, "POP3 Extension Mechanism", RFC 2449, November 1998.
- [RFC3454] Hoffman, P. and M. Blanchet, "Preparation of Internationalized Strings ("stringprep")", RFC 3454, December 2002.
- [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.
- [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.
- [RFC6530] Klensin, J. and Y. Ko, "Overview and Framework for Internationalized Email", RFC 6530, February 2012.
- [RFC6532] Yang, A., Steele, S., and N. Freed, "Internationalized Email Headers", RFC 6532, February 2012.

9.2. Informative References

- [RFC2595] Newman, C., "Using TLS with IMAP, POP3 and ACAP", RFC 2595, June 1999.
- [RFC5034] Siemborski, R. and A. Menon-Sen, "The Post Office Protocol (POP3) Simple Authentication and Security Layer (SASL) Authentication Mechanism", RFC 5034, July 2007.
- [RFC5721] Gellens, R. and C. Newman, "POP3 Support for UTF-8", RFC 5721, February 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.

Appendix B. Acknowledgments

Thanks to John Klensin, Joseph Yee, Tony Hansen, Alexey Melnikov 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: Proposed Standard
Updates: 3501

Arnt Gulbrandsen
June 2012

EAI: Simplified POP/IMAP downgrading
draft-ietf-eai-simplifiedowngrade-05.txt

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>. The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft expires in November 2012.

Abstract

This document specifies a method for IMAP and POP servers to serve internationalized messages to conventional clients. The specification is simple, easy to implement and provides only rudimentary results.

1. Overview

It may happen that a conventional IMAP or POP client opens a mailbox containing internationalized messages, or even attempt to read internationalized messages, for instance when a user has both internationalized and conventional MUAs.

Some operations cannot be performed by conventional clients. Most importantly, one or more addresses on an internationalized message are not valid according to the specifications the client uses, so address-based operations are not possible. This includes displaying the addresses, replying, and most types of address-based signature or security processing.

Still, the sender's name, the message subject, body text and attachments can easily be displayed, so a helpful IMAP/POP server may prefer to provide access to what it can rather than hide the message entirely.

This document specifies a way to present such messages to the client. It values simplicity of implementation over fidelity of representation, since implementing a high-fidelity downgrade algorithm is likely more work than implementing proper support for [RFC5721] and/or [RFC5738].

The server is assumed to be internationalized internally, and to store messages internationalized messages natively. When it needs to present an internationalized message to a conventional client, it synthesizes a conventional message containing most of the information and presents that (the "synthetic message").

2. Information preserved and lost

The synthetic message is intended to convey the most important information to the user. Where information is lost, the user should see the message as incomplete rather than modified.

The synthetic message is not intended to convey any information to the client software.

Upper case in examples represents non-ASCII. example.com is a plain domain, EXAMPLE.com represents a non-ASCII .com domain.

2.1 Email addresses

Each internationalized email address in the header fields listed below is replaced with an invalid email address whose display-name tells the user what happened.

The format of the display-name is explicitly unspecified. Anything which tells the user what happened is good. Anything which produces an email address which might belong to someone else is bad.

Given an internationalized address "Fred Foo <fred@EXAMPLE.com>", an implementation may choose to render it e.g. as these examples:

```
"fred@EXAMPLE.com" <invalid@internationalized-address.invalid>
Fred Foo <invalid@internationalized.invalid>
internationalized-address:;
fred:;
```

(The .invalid top-level domain is reserved by [RFC2606], therefore the first two examples are syntactically valid, but will never belong to anyone. Note that the display-name often will need [RFC2047] encoding.)

The affected header fields are Bcc, Cc, From, Reply-To, Resent-Bcc, Resent-Cc, Resent-From, Resent-Sender, Resent-To, Return-Path, Sender and To. Any addresses present in other header fields are not regarded as addresses by this specification.

2.2 MIME parameters

Any MIME parameter [RFC2045] (whether in the message header or a bodypart header) which cannot be presented as-is to the client is silently excised.

Given a field such as

```
Content-Disposition: attachment; filename=FOO
```

the field is presented as

```
Content-Disposition: attachment
```

2.3 "Subject"

If the Subject field cannot be presented as-is, the server presents a representation encoded as specified in [RFC2047].

2.4 Remaining header fields

Any header field which cannot be presented to the client even after the modifications in sections 2.1-2.3 is silently excised.

3. IMAP-specific details

IMAP allows clients to retrieve the message size without downloading it, using RFC822.SIZE, BODY.SIZE[] and so on. [RFC3501] requires that the returned size be exact.

This specification relaxes that requirement: When a conventional client requests size information for a message, the IMAP server is permitted to return size information for the internationalized message, even though the synthetic message's size differs.

When an IMAP server carries out downgrading as part of generating nFETCH responses, it reports which messages were synthesised using a response code and attendant UID set. This can be helpful to humans debugging the server and/or client.

```
C: a UID FETCH 1:* BODY.PEEK[HEADER.FIELDS(To From Cc)]
S: 1 FETCH (UID 65 [...])
S: 2 FETCH (UID 70 [...])
S: a OK [DOWNGRADED 70,105,108,109] Done
```

The message-set argument to DOWNGRADED contains UIDs.

Note that DOWNGRADED does not necessarily mention all the internationalized messages in the mailbox. In the example above, we know that UID 65 does not contain internationalized addresses in From, To and Cc. It may contain an internationalized Subject, etc.

4. POP-specific details

The number of lines specified in the TOP command (see [RFC1939]) refers to the synthetic message. The message size reported by e.g. LIST may refer to either the internationalized or the synthetic message.

5. Security Considerations

If the internationalized message contains signed body parts, the synthetic message probably contains an invalid signature. This is a necessary limitation of displaying internationalized messages in conventional clients, since the client does not support internationalized addresses.

If any excised information is significant, then that information does not arrive at the recipient. Notably, the message-id, in-reference-to and/or references fields may be excised, which might cause a lack of context when the recipient reads the message.

Some POP/IMAP clients delete the original message and use only the what they downloaded, Fetchmail is one well-known example. This may lead to permanent loss of information.

Other clients cache messages for a very long time, even across client upgrades, such as the stock Android client. When such a client is internationalized, care must be taken so that it will not use an old synthetic message from its cache rather than retrieve the real message from the server.

6. Acknowledgements

Claudio Allocchio, Ned Freed, Kazunori Fujiwara, Ted Hardie, Barry Leiba, John Levine, Alexey Melnikov, Chris Newman, Joseph Yee and the originator of rule 12 in [RFC1925] helped with this document.

7. IANA Considerations

The IANA is requested to add DOWNGRADED to the IMAP response code registry.

(RFC editor: Please remove this paragraph. I can't remember the URL of the registry, but it's the one specified in RFC 5530.)

8. Normative References

- [RFC1939] Myers, J and M. Rose, "Post Office Protocol - Version 3", RFC 1939, Carnegie Mellon, 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, "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, University of Tennessee, November 1996.
- [RFC2606] Eastlake, D. and A. Panitz, "Reserved Top Level DNS Names", BCP 32, RFC 2606, June 1999.
- [RFC3501] Crispin, "Internet Message Access Protocol - Version 4rev1", RFC 3501, University of Washington, June 2003.

9. Informative References

- [RFC1925] Callon, R., "Fundamental Truths of Networking", RFC 1925, Bay Networks, April 1996.
- [RFC5721] Gellens, R., and C. Newman, "POP3 Support for UTF-8", RFC 5721, Qualcomm Incorporated, February 2010.
- [RFC5738] Resnick, P. and C. Newman, "IMAP Support for UTF-8", RFC 5738, Qualcomm Incorporated, March 2010.

10. Author's Address

Arnt Gulbrandsen
Schweppermannstr. 8
D-81671 Muenchen
Germany

Fax: +49 89 4502 9758

Email: arnt@gulbrandsen.priv.no

(RFC Editor: Please delete everything after this point)

Open Issues

Should Kazunori Fujiwara's downgrade document also mention DOWNGRADED?

RFC Editor: IF 5721 and/or 5738 have been superseded by new RFCs at this time, please change the references to those RFCs throughout this document. Well, except in the previous sentence. I'm such a pedant.

Changes since -00

Added a rule to handle Subject

Removed the sentence about unknown;;

Terminology fixes

Changes since -01

Nits from Joseph Yee.

Clarified the address rendering and added non-.invalid examples, based on suggestions from Kazunori Fujiwara.

Many changes from Barry Leiba: Simplified and better terminology, reformatted examples, more references, etc.

Specified POP TOP. A bit of a no-op specification.

Mention BODY.SIZE[] as well as RFC822.SIZE. Wave hands so BODY.SIZE[1] sneaks past.

<http://rant.gulbrandsen.priv.no/good-bad-rfc> fwiw

Changes since -02

Added the DOWNGRADED response code, since both Barry and Alexey wants it.

Changes since -03

Added/changed text in response to appsdireviews from Ted Hardie and Claudio Allocchio.

Changes since -04

Closed two open issues; the interest in them was clearly negligible.

"Updates: 3501" because of the SIZE relaxation.

Security considerations about download-and-delete and long-term caching.

Bring on the WGLC!