

Network Working Group  
Request for Comments: 5337  
Updates: 3461, 3464, 3798  
Category: Experimental

C. Newman  
Sun Microsystems  
A. Melnikov, Ed.  
Isode Ltd  
September 2008

## Internationalized Delivery Status and Disposition Notifications

### Status of This Memo

This memo defines an Experimental Protocol for the Internet community. It does not specify an Internet standard of any kind. Discussion and suggestions for improvement are requested. Distribution of this memo is unlimited.

### Abstract

Delivery status notifications (DSNs) are critical to the correct operation of an email system. However, the existing Draft Standards (RFC 3461, RFC 3462, RFC 3464) are presently limited to US-ASCII text in the machine-readable portions of the protocol. This specification adds a new address type for international email addresses so an original recipient address with non-US-ASCII characters can be correctly preserved even after downgrading. This also provides updated content return media types for delivery status notifications and message disposition notifications to support use of the new address type.

This document experimentally extends RFC 3461, RFC 3464, and RFC 3798.

Table of Contents

- 1. Introduction . . . . . 3
- 2. Conventions Used in This Document . . . . . 3
- 3. UTF-8 Address Type . . . . . 3
- 4. UTF-8 Delivery Status Notifications . . . . . 6
  - 4.1. Additional Requirements on SMTP Servers . . . . . 8
- 5. UTF-8 Message Disposition Notifications . . . . . 9
- 6. IANA Considerations . . . . . 10
  - 6.1. UTF-8 Mail Address Type Registration . . . . . 10
  - 6.2. Update to 'smtp' Diagnostic Type Registration . . . . . 11
  - 6.3. message/global-headers . . . . . 11
  - 6.4. message/global-delivery-status . . . . . 12
  - 6.5. message/global-disposition-notification . . . . . 13
- 7. Security Considerations . . . . . 15
- 8. References . . . . . 15
  - 8.1. Normative References . . . . . 15
  - 8.2. Informative References . . . . . 16
- Appendix A. Acknowledgements . . . . . 17

## 1. Introduction

When an email message is transmitted using the UTF8SMTP [RFC5336] extension and Internationalized Email Headers [RFC5335], it is sometimes necessary to return that message or generate a Message Disposition Notification (MDN) [RFC3798]. As a message sent to multiple recipients can generate a status and disposition notification for each recipient, it is helpful if a client can correlate these notifications based on the recipient address it provided; thus, preservation of the original recipient is important. This specification describes how to preserve the original recipient and updates the MDN and DSN formats to support the new address types.

## 2. Conventions Used in This Document

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

The formal syntax use the Augmented Backus-Naur Form (ABNF) [RFC5234] notation including the core rules defined in Appendix B of RFC 5234 [RFC5234] and the UTF-8 syntax rules in Section 4 of [RFC3629].

## 3. UTF-8 Address Type

An Extensible Message Format for Delivery Status Notifications [RFC3464] defines the concept of an address type. The address format introduced in Internationalized Email Headers [RFC5335] is a new address type. The syntax for the new address type in the context of status notifications is specified at the end of this section.

An SMTP [RFC2821] server that advertises both the UTF8SMTP extension [RFC5336] and the DSN extension [RFC3461] MUST accept a UTF-8 address type in the ORCPT parameter including 8-bit UTF-8 characters. This address type also includes a 7-bit encoding suitable for use in a message/delivery-status body part or an ORCPT parameter sent to an SMTP server that does not advertise UTF8SMTP.

This address type has 3 forms: utf-8-addr-xtext, utf-8-addr-unitext, and utf-8-address. The first 2 forms are 7-bit safe.

The utf-8-address form is only suitable for use in newly defined protocols capable of native representation of 8-bit characters. That is, the utf-8-address form MUST NOT be used in the ORCPT parameter when the SMTP server doesn't advertise support for UTF8SMTP or the SMTP server supports UTF8SMTP, but the address contains US-ASCII characters not permitted in the ORCPT parameter (e.g., the ORCPT parameter forbids unencoded SP and the = character), or in a 7-bit

transport environment including a message/global-delivery-status Original-Recipient or Final-Recipient field. In the former case, the utf-8-addr-xttext form (see below) MUST be used instead; in the latter case, the utf-8-addr-unitext form MUST be used. The utf-8-address form MAY be used in the ORCPT parameter when the SMTP server also advertises support for UTF8SMTP and the address doesn't contain any US-ASCII characters not permitted in the ORCPT parameter. It SHOULD be used in a message/global-delivery-status Original-Recipient or Final-Recipient DSN field, or in an Original-Recipient header field [RFC3798] if the message is a UTF8SMTP message.

In addition, the utf-8-addr-unitext form can be used anywhere where the utf-8-address form is allowed.

When using in the ORCPT parameter, the UTF-8 address type requires that US-ASCII CTLs, SP, \, +, and = be encoded using xttext encoding as described in [RFC3461]. This is described by the utf-8-addr-xttext form in the ABNF below. Unicode characters MAY be included in a UTF-8 address type using a "\x{HEXPOINT}" syntax (EmbeddedUnicodeChar), where HEXPOINT is 2 to 6 hexadecimal digits. When sending data to a UTF8SMTP-capable server, native UTF-8 characters SHOULD be used instead of the EmbeddedUnicodeChar syntax described in details below. When sending data to an SMTP server that does not advertise UTF8SMTP, then the EmbeddedUnicodeChar syntax MUST be used instead of UTF-8.

When the ORCPT parameter is placed in a message/global-delivery-status Original-Recipient field, the utf-8-addr-xttext form of the UTF-8 address type SHOULD be converted to the utf-8-address form (see the ABNF below) by removing all xttext encoding first (which will result in the utf-8-addr-unitext form), followed by removal of the unitext encoding. However, if an address is labeled with the UTF-8 address type but does not conform to utf-8 syntax, then it MUST be copied into the message/global-delivery-status field without alteration.

The ability to encode characters with the EmbeddedUnicodeChar encodings should be viewed as a transitional mechanism. It is hoped that as systems lacking support for UTF8SMTP become less common over time, these encodings can eventually be phased out.

In the ABNF below, all productions not defined in this document are defined in Appendix B of [RFC5234], in Section 4 of [RFC3629], or in [RFC3464].

```

utf-8-type-addr      = "utf-8;" utf-8-enc-addr

utf-8-address        = uMailbox [ 1*WSP "<" Mailbox ">" ]
; uMailbox is defined in [RFC5336].
; Mailbox is defined in [RFC2821].

utf-8-enc-addr       = utf-8-addr-xtext /
                      utf-8-addr-unitext /
                      utf-8-address

utf-8-addr-xtext     = xtext
; xtext is defined in [RFC3461].
; When xtext encoding is removed,
; the syntax MUST conform to
; utf-8-addr-unitext.

utf-8-addr-unitext   = 1*(QUCHAR / EmbeddedUnicodeChar)
; MUST follow utf-8-address ABNF when
; dequoted

QUCHAR                = %x21-2a / %x2c-3c / %x3e-5b / %x5d-7e /
                      UTF8-2 / UTF8-3 / UTF8-4
; US-ASCII printable characters except
; CTLs, SP, '\', '+' and '=', plus
; other Unicode characters in UTF-8

EmbeddedUnicodeChar  = %x5C.78 "{" HEXPOINT "}"
; starts with "\x"

HEXPOINT = "5C" / (HEXDIG8 HEXDIG) /      ; 2 digit forms
           ( NZHEXDIG 2(HEXDIG) ) /      ; 3 digit forms
           ( NZDHEXDIG 3(HEXDIG) ) /
           ( "D" %x30-37 2(HEXDIG) ) /
           ; 4 digit forms excluding surrogate
           ( NZHEXDIG 4(HEXDIG) ) /      ; 5 digit forms
           ( "10" 4*HEXDIG )           ; 6 digit forms
; represents either "\" or a Unicode code point outside the
; US-ASCII repertoire

HEXDIG8                = %x38-39 / "A" / "B" / "C" / "D" / "E" / "F"
; HEXDIG excluding 0-7

NZHEXDIG               = %x31-39 / "A" / "B" / "C" / "D" / "E" / "F"
; HEXDIG excluding "0"

NZDHEXDIG              = %x31-39 / "A" / "B" / "C" / "E" / "F"
; HEXDIG excluding "0" and "D"

```

#### 4. UTF-8 Delivery Status Notifications

A traditional delivery status notification [RFC3464] comes in a three-part multipart/report [RFC3462] container, where the first part is human-readable text describing the error, the second part is a 7-bit-only message/delivery-status, and the optional third part is used for content (message/rfc822) or header (text/rfc822-headers) return. As the present DSN format does not permit returning of undeliverable UTF8SMTP messages, three new media types are needed.

The first type, message/global-delivery-status, has the syntax of message/delivery-status with three modifications. First, the charset for message/global-delivery-status is UTF-8, and thus any field MAY contain UTF-8 characters when appropriate (see the ABNF below). In particular, the Diagnostic-Code field MAY contain UTF-8 as described in UTF8SMTP [RFC5336]; the Diagnostic-Code field SHOULD be in i-default language [DEFAULTLANG]. Second, systems generating a message/global-delivery-status body part SHOULD use the utf-8-address form of the UTF-8 address type for all addresses containing characters outside the US-ASCII repertoire. These systems SHOULD up-convert the utf-8-addr-xtext or the utf-8-addr-unitext form of a UTF-8 address type in the ORCPT parameter to the utf-8-address form of a UTF-8 address type in the Original-Recipient field. Third, a new optional field called Localized-Diagnostic is added. Each instance includes a language tag [LANGTAGS] and contains text in the specified language. This is equivalent to the text part of the Diagnostic-Code field. All instances of Localized-Diagnostic MUST use different language tags. The ABNF for message/global-delivery-status is specified below.

In the ABNF below, all productions not defined in this document are defined in Appendix B of [RFC5234], in Section 4 of [RFC3629], or in [RFC3464].

```
utf-8-delivery-status-content = per-message-fields
                               1*( CRLF utf-8-per-recipient-fields )
; "per-message-fields" remains unchanged from the definition
; in RFC 3464, except for the "extension-field"
; which is updated below.
```

```
utf-8-per-recipient-fields =
  [ original-recipient-field CRLF ]
  final-recipient-field CRLF
  action-field CRLF
  status-field CRLF
  [ remote-mta-field CRLF ]
  [ diagnostic-code-field CRLF
    *(localized-diagnostic-text-field CRLF) ]
  [ last-attempt-date-field CRLF ]
  [ will-retry-until-field CRLF ]
  *( extension-field CRLF )
; All fields except for "original-recipient-field",
; "final-recipient-field", "diagnostic-code-field"
; and "extension-field" remain unchanged from
; the definition in RFC 3464.

generic-address =/ utf-8-enc-addr
; Only allowed with the "utf-8" address-type.
;
; This indirectly updates "original-recipient-field"
; and "final-recipient-field"

diagnostic-code-field =
  "Diagnostic-Code" ":" diagnostic-type ";" *text-fixed

localized-diagnostic-text-field =
  "Localized-Diagnostic" ":" Language-Tag ";" *utf8-text
; "Language-Tag" is a language tag as defined in [LANGTAGS].

extension-field =/ extension-field-name ":" *utf8-text

text-fixed = %d1-9 /          ; Any Unicode character except for NUL,
               %d11 /         ; CR and LF, encoded in UTF-8
               %d12 /
               %d14-127
; Same as <text> from [RFC2822], but without <obs-text>.
; If/when RFC 2822 is updated to disallow <obs-text>,
; this should become just <text>
; Also, if/when RFC 2822 is updated to disallow control characters
; this should become a reference to RFC 2822upd instead.

utf8-text = text-fixed / UTF8-non-ascii

UTF8-non-ascii = UTF8-2 / UTF8-3 / UTF8-4
```

The second type, used for returning the content, is message/global which is similar to message/rfc822, except it contains a message with UTF-8 headers. This media type is described in [RFC5335].

The third type, used for returning the headers, is message/global-headers and contains only the UTF-8 header fields of a message (all lines prior to the first blank line in a UTF8SMTP message). Unlike message/global, this body part provides no difficulties for the present infrastructure.

Note that as far as multipart/report [RFC3462] container is concerned, message/global-delivery-status, message/global, and message/global-headers MUST be treated as equivalent to message/delivery-status, message/rfc822, and text/rfc822-headers. That is, implementations processing multipart/report MUST expect any combinations of the 6 MIME types mentioned above inside a multipart/report MIME type.

All three new types will typically use the "8bit" Content-Transfer-Encoding. (In the event all content is 7-bit, the equivalent traditional types for delivery status notifications MAY be used. For example, if information in message/global-delivery-status part can be represented without any loss of information as message/delivery-status, then the message/delivery-status body part may be used.) Note that [RFC5335] relaxed restriction from MIME [RFC2046] regarding use of Content-Transfer-Encoding in new "message" subtypes. This specification explicitly allows use of Content-Transfer-Encoding in message/global-headers and message/global-delivery-status. This is not believed to be problematic as these new MIME types are intended primarily for use by newer systems with full support for 8-bit MIME and UTF-8 headers.

#### 4.1. Additional Requirements on SMTP Servers

If an SMTP server that advertises both UTF8SMTP and DSN needs to return an undeliverable UTF8SMTP message, then it MUST NOT downgrade [DOWNGRADE] the UTF8SMTP message when generating the corresponding multipart/report. If the return path SMTP server does not support UTF8SMTP, then the undeliverable body part and headers MUST be encoded using a 7-bit Content-Transfer-Encoding such as "base64" or "quoted-printable" [RFC2045], as detailed in Section 4. Otherwise, "8bit" Content-Transfer-Encoding can be used.

## 5. UTF-8 Message Disposition Notifications

Message Disposition Notifications [RFC3798] have a similar design and structure to DSNs. As a result, they use the same basic return format. When generating an MDN for a UTF-8 header message, the third part of the multipart/report contains the returned content (message/global) or header (message/global-headers), same as for DSNs. The second part of the multipart/report uses a new media type, message/global-disposition-notification, which has the syntax of message/global-disposition-notification with two modifications. First, the charset for message/global-disposition-notification is UTF-8, and thus any field MAY contain UTF-8 characters when appropriate (see the ABNF below). (In particular, the failure-field, the error-field, and the warning-field MAY contain UTF-8. These fields SHOULD be in i-default language [DEFAULTLANG].) Second, systems generating a message/global-disposition-notification body part (typically a mail user agent) SHOULD use the UTF-8 address type for all addresses containing characters outside the US-ASCII repertoire.

The MDN specification also defines the Original-Recipient header field, which is added with a copy of the contents of ORCPT at delivery time. When generating an Original-Recipient header field, a delivery agent writing a UTF-8 header message in native format SHOULD convert the utf-8-addr-xttext or the utf-8-addr-unitext form of a UTF-8 address type in the ORCPT parameter to the corresponding utf-8-address form.

The MDN specification also defines the Disposition-Notification-To header, which is an address header and thus follows the same 8-bit rules as other address headers such as "From" and "To" when used in a UTF-8 header message.

```
; ABNF for "original-recipient-header", "original-recipient-field",  
; and "final-recipient-field" from RFC 3798 is implicitly updated  
; as they use the updated "generic-address" as defined in  
; Section 4 of this document.
```

```
failure-field = "Failure" ":" *utf8-text  
; "utf8-text" is defined in Section 4 of this document.
```

```
error-field = "Error" ":" *utf8-text  
; "utf8-text" is defined in Section 4 of this document.
```

```
warning-field = "Warning" ":" *utf8-text  
; "utf8-text" is defined in Section 4 of this document.
```

## 6. IANA Considerations

This specification does not create any new IANA registries. However, the following items have been registered as a result of this document.

### 6.1. UTF-8 Mail Address Type Registration

The mail address type registry was created by RFC 3464. The registration template response follows:

- (a) The proposed address-type name.

UTF-8

- (b) The syntax for mailbox addresses of this type, specified using BNF, regular expressions, ASN.1, or other non-ambiguous language.

See Section 3.

- (c) If addresses of this type are not composed entirely of graphic characters from the US-ASCII repertoire, a specification for how they are to be encoded as graphic US-ASCII characters in a DSN Original-Recipient or Final-Recipient DSN field.

This address type has 3 forms (as defined in Section 3): `utf-8-addr-xtext`, `utf-8-addr-unitext`, and `utf-8-address`. The first 2 forms are 7-bit safe.

The `utf-8-address` form MUST NOT be used

1. in the ORCPT parameter when the SMTP server doesn't advertise support for UTF8SMTP;
2. or the SMTP server supports UTF8SMTP, but the address contains US-ASCII characters not permitted in the ORCPT parameter (e.g., the ORCPT parameter forbids SP and the = characters);
3. or in a 7-bit transport environment including a message/delivery-status Original-Recipient or Final-Recipient field.

The `utf-8-addr-xtext` form MUST be used instead in the first case; the `utf-8-addr-unitext` form MUST be used in the other two cases. The `utf-8-address` form MAY be used in the ORCPT parameter when the SMTP server also advertises support for UTF8SMTP and the address doesn't contain any US-ASCII characters not permitted in the ORCPT parameter;

in a message/global-delivery-status Original-Recipient or Final-Recipient DSN field; or in an Original-Recipient header field [RFC3798] if the message is a UTF8SMTP message.

In addition, the utf-8-addr-unitext form can be used anywhere where the utf-8-address form is allowed.

## 6.2. Update to 'smtp' Diagnostic Type Registration

The mail diagnostic type registry was created by RFC 3464. The registration for the 'smtp' diagnostic type should be updated to reference RFC 5337 in addition to RFC 3464.

When the 'smtp' diagnostic type is used in the context of a message/global-delivery-status body part, it remains as presently defined. When the 'smtp' diagnostic type is used in the context of a message/global-delivery-status body part, the codes remain the same, but the text portion MAY contain UTF-8 characters.

## 6.3. message/global-headers

Type name: message

Subtype name: global-headers

Required parameters: none

Optional parameters: none

Encoding considerations: This media type contains Internationalized Email Headers [RFC5335] with no message body. Whenever possible, the 8-bit content transfer encoding SHOULD be used. When this media type passes through a 7-bit-only SMTP infrastructure it MAY be encoded with the base64 or quoted-printable content transfer encoding.

Security considerations: See Section 7.

Interoperability considerations: It is important that this media type is not converted to a charset other than UTF-8. As a result, implementations MUST NOT include a charset parameter with this media type. Although it might be possible to downconvert this media type to the text/rfc822-header media type, such conversion is discouraged as it loses information.

Published specification: RFC 5337

Applications that use this media type: UTF8SMTP servers and email clients that support multipart/report generation or parsing.

Additional information:

Magic number(s): none

File extension(s): In the event this is saved to a file, the extension ".u8hdr" is suggested.

Macintosh file type code(s): The 'TEXT' type code is suggested as files of this type are typically used for diagnostic purposes and suitable for analysis in a UTF-8 aware text editor. A uniform type identifier (UTI) of "public.utf8-email-message-header" is suggested. This type conforms to "public.utf8-plain-text" and "public.plain-text".

Person & email address to contact for further information: See the Authors' Addresses section of this document.

Intended usage: COMMON

Restrictions on usage: This media type contains textual data in the UTF-8 charset. It typically contains octets with the 8th bit set. As a result, a transfer encoding is required when a 7-bit transport is used.

Author: See the Authors' Addresses section of this document.

Change controller: IETF Standards Process

#### 6.4. message/global-delivery-status

Type name: message

Subtype name: global-delivery-status

Required parameters: none

Optional parameters: none

Encoding considerations: This media type contains delivery status notification attributes in the UTF-8 charset. The 8-bit content transfer encoding MUST be used with this content-type, unless it is sent over a 7-bit transport environment in which case quoted-printable or base64 may be necessary.

Security considerations: See Section 7

Interoperability considerations: This media type provides functionality similar to the message/delivery-status content-type for email message return information. Clients of the previous format will need to be upgraded to interpret the new format; however, the new media type makes it simple to identify the difference.

Published specification: RFC 5337

Applications that use this media type: SMTP servers and email clients that support delivery status notification generation or parsing.

Additional information:

Magic number(s): none

File extension(s): The extension ".u8dsn" is suggested.

Macintosh file type code(s): A uniform type identifier (UTI) of "public.utf8-email-message-delivery-status" is suggested. This type conforms to "public.utf8-plain-text".

Person & email address to contact for further information: See the Authors' Addresses section of this document.

Intended usage: COMMON

Restrictions on usage: This is expected to be the second part of a multipart/report.

Author: See the Authors' Addresses section of this document.

Change controller: IETF Standards Process

#### 6.5. message/global-disposition-notification

Type name: message

Subtype name: global-disposition-notification

Required parameters: none

Optional parameters: none

Encoding considerations: This media type contains disposition notification attributes in the UTF-8 charset. The 8-bit content transfer encoding **MUST** be used with this content-type, unless it is sent over a 7-bit transport environment in which case quoted-printable or base64 may be necessary.

Security considerations: See Section 7.

Interoperability considerations: This media type provides functionality similar to the message/disposition-notification content-type for email message disposition information. Clients of the previous format will need to be upgraded to interpret the new format; however, the new media type makes it simple to identify the difference.

Published specification: RFC 5337

Applications that use this media type: Email clients or servers that support message disposition notification generation or parsing.

Additional information:

Magic number(s): none

File extension(s): The extension ".u8mdn" is suggested.

Macintosh file type code(s): A uniform type identifier (UTI) of "public.utf8-email-message-disposition-notification" is suggested. This type conforms to "public.utf8-plain-text".

Person & email address to contact for further information: See the Authors' Addresses section of this document.

Intended usage: COMMON

Restrictions on usage: This is expected to be the second part of a multipart/report.

Author: See the Authors' Addresses section of this document.

Change controller: IETF Standards Process

## 7. Security Considerations

Automated use of report types without authentication presents several security issues. Forging negative reports presents the opportunity for denial-of-service attacks when the reports are used for automated maintenance of directories or mailing lists. Forging positive reports may cause the sender to incorrectly believe a message was delivered when it was not.

Malicious users can generate report structures designed to trigger coding flaws in report parsers. Report parsers need to use secure coding techniques to avoid the risk of buffer overflow or denial-of-service attacks against parser coding mistakes. Code reviews of such parsers are also recommended.

Malicious users of the email system regularly send messages with forged envelope return paths, and these messages trigger delivery status reports that result in a large amount of unwanted traffic on the Internet. Many users choose to ignore delivery status notifications because they are usually the result of "blowback" from forged messages and thus never notice when messages they sent go undelivered. As a result, support for correlation of delivery status and message disposition notification messages with sent-messages has become a critical feature of mail clients and possibly mail stores if the email infrastructure is to remain reliable. In the short term, simply correlating message-IDs may be sufficient to distinguish true status notifications from those resulting from forged originator addresses. But in the longer term, including cryptographic signature material that can securely associate the status notification with the original message is advisable.

As this specification permits UTF-8 in additional fields, the security considerations of UTF-8 [RFC3629] apply.

## 8. References

### 8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2821] Klensin, J., "Simple Mail Transfer Protocol", RFC 2821, April 2001.
- [RFC2822] Resnick, P., "Internet Message Format", RFC 2822, April 2001.

- [RFC3461] Moore, K., "Simple Mail Transfer Protocol (SMTP) Service Extension for Delivery Status Notifications (DSNs)", RFC 3461, January 2003.
- [RFC3462] Vaudreuil, G., "The Multipart/Report Content Type for the Reporting of Mail System Administrative Messages", RFC 3462, January 2003.
- [RFC3464] Moore, K. and G. Vaudreuil, "An Extensible Message Format for Delivery Status Notifications", RFC 3464, January 2003.
- [RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.
- [RFC3798] Hansen, T. and G. Vaudreuil, "Message Disposition Notification", RFC 3798, May 2004.
- [RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.
- [RFC5335] Yang, A., Ed., "Internationalized Email Headers", RFC 5335, September 2008.
- [RFC5336] Yao, J., Ed. and W. Mao, Ed., "SMTP Extension for Internationalized Email Addresses", RFC 5336, September 2008.
- [LANGTAGS] Phillips, A. and M. Davis, "Tags for Identifying Languages", RFC 4646, September 2006.
- [DEFAULTLANG] Alvestrand, H., "IETF Policy on Character Sets and Languages", RFC 2277, January 1998.

## 8.2. Informative References

- [RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.
- [RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, November 1996.
- [DOWNGRADE] Fujiwara, K. and Y. Yoneya, "Downgrading mechanism for Email Address Internationalization", Work in Progress, July 2008.

## Appendix A. Acknowledgements

Many thanks for input provided by Pete Resnick, James Galvin, Ned Freed, John Klensin, Harald Alvestrand, Frank Ellermann, SM, and members of the EAI WG to help solidify this proposal.

### Authors' Addresses

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

EMail: [chris.newman@sun.com](mailto:chris.newman@sun.com)

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

EMail: [Alexey.Melnikov@isode.com](mailto:Alexey.Melnikov@isode.com)

## Full Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

