

Internet Engineering Task Force (IETF)  
Request for Comments: 6533  
Obsoletes: 5337  
Updates: 3461, 3464, 3798, 6522  
Category: Standards Track  
ISSN: 2070-1721

T. Hansen, Ed.  
AT&T Laboratories  
C. Newman  
Oracle  
A. Melnikov  
Isode Ltd  
February 2012

## Internationalized Delivery Status and Disposition Notifications

### Abstract

Delivery status notifications (DSNs) are critical to the correct operation of an email system. However, the existing Draft Standards (RFC 3461, RFC 3464, RFC 6522) are presently limited to 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-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 extends RFC 3461, RFC 3464, RFC 3798, and RFC 6522.

### Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc6533>.

## 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. UTF-8 Address Type . . . . .	3
4. UTF-8 Delivery Status Notifications . . . . .	6
4.1. The message/global-delivery-status Media Type . . . . .	6
4.2. The message/global Media Type . . . . .	8
4.3. The message/global-headers Media Type . . . . .	8
4.4. Using These Media Types with multipart/report . . . . .	8
4.5. Additional Requirements on SMTP Servers . . . . .	9
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 . . . . .	14
7. Security Considerations . . . . .	15
8. References . . . . .	16
8.1. Normative References . . . . .	16
8.2. Informative References . . . . .	17
Appendix A. Changes since RFC 5337 . . . . .	18
Appendix B. Acknowledgements . . . . .	18

## 1. Introduction

When an email message is transmitted using the SMTPUTF8 [RFC6531] extension and Internationalized Email Headers [RFC6532], 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.

NOTE: While this specification updates the experimental versions of this protocol by removing certain constructs (e.g., the "<addr <addr>>" address syntax is no longer permitted), the name of the Address Type "UTF-8" and the media type names message/global, message/global-delivery-status, and message/global-headers have not been changed.

This specification is a revision of and replacement for [RFC5337]. Section 6 of [RFC6530] describes the change in approach between this specification and the previous version.

## 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 uses the Augmented Backus-Naur Form (ABNF) [RFC5234] notation including the core rules defined in Appendix B of [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" [RFC6532] 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 [RFC5321] server that advertises both the SMTPUTF8 extension [RFC6531] 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 SMTPUTF8.

This address type has 3 forms: `utf-8-addr-xtext`, `utf-8-addr-unixtext`, and `utf-8-address`. Only the first form is 7-bit safe (only uses ASCII characters [ASCII]).

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:

1. in the ORCPT parameter when the SMTP server doesn't advertise support for SMTPUTF8 (`utf-8-addr-xtext` **MUST** be used instead); or
2. if the SMTP server supports SMTPUTF8, but the address contains ASCII characters not permitted in the ORCPT parameter (e.g., the ORCPT parameter forbids unencoded SP and the '=' character), (either `utf-8-addr-unixtext` or `utf-8-addr-xtext` **MUST** be used instead); or
3. in a 7-bit transport environment including a message/delivery-status "Original-Recipient:" or "Final-Recipient:" field, (`utf-8-addr-xtext` **MUST** be used instead).

The `utf-8-address` form **MAY** be used in the ORCPT parameter when the SMTP server also advertises support for SMTPUTF8 and the address doesn't contain any 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 SMTPUTF8 message.

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

When used in the ORCPT parameter, the UTF-8 address type requires that ASCII CTLs, SP, '\', '+', and '=' be encoded using 'unixtext' encoding (see below). This is described by the `utf-8-addr-xtext` and `utf-8-addr-unixtext` forms in the ABNF below. The 'unixtext' encoding uses "\x{HEXPOINT}" syntax (EmbeddedUnicodeChar in the ABNF below) for encoding any Unicode character outside of ASCII range, as well as for encoding CTLs, SP, '\', '+', and '='. HEXPOINT is 2 to 6 hexadecimal digits. This encoding avoids the need to use the xtext encoding described in [RFC3461], as any ASCII characters that need to be escaped using xtext encoding never appear in any unixtext-encoded string. When sending data to a SMTPUTF8-capable server, native UTF-8 characters **SHOULD** be used instead of the EmbeddedUnicodeChar syntax described below. When sending data to an SMTP server that does not advertise SMTPUTF8, 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-xtext form of the UTF-8 address type SHOULD be converted to the utf-8-address form (see the ABNF below) by removing 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 and avoided when possible. It is hoped that as systems lacking support for SMTPUTF8 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        = Mailbox
                      ; Mailbox as defined in [RFC6531].

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

utf-8-addr-xtext     = 1*(QCHAR / EmbeddedUnicodeChar)
                      ; 7bit form of utf-8-addr-unitext.
                      ; Safe for use in the ORCPT [RFC3461]
                      ; parameter even when SMTPUTF8 SMTP
                      ; extension is not advertised.

utf-8-addr-unitext   = 1*(QUCHAR / EmbeddedUnicodeChar)
                      ; MUST follow utf-8-address ABNF when
                      ; dequoted.
                      ; Safe for using in the ORCPT [RFC3461]
                      ; parameter when SMTPUTF8 SMTP extension
                      ; is also advertised.

QCHAR                = %x21-2a / %x2c-3c / %x3e-5b / %x5d-7e
                      ; ASCII printable characters except
                      ; CTLs, SP, '\', '+', '='.
```

QUCHAR = QCHAR / UTF8-2 / UTF8-3 / UTF8-4  
 ; ASCII printable characters except  
 ; CTLs, SP, '\', '+' and '=', plus  
 ; other Unicode characters encoded in UTF-8

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

HEXPOINT = ( ( "0"/"1" ) %x31-39 ) / "10" / "20" /  
 "2B" / "3D" / "7F" / ; all xtext-specials  
 "5C" / (HEXDIG8 HEXDIG) / ; 2-digit forms  
 ( NZHEXDIG 2(HEXDIG) ) / ; 3-digit forms  
 ( NZDHEXDIG 3(HEXDIG) ) / ; 4-digit forms excluding  
 ( "D" %x30-37 2(HEXDIG) ) / ; ... surrogate  
 ( NZHEXDIG 4(HEXDIG) ) / ; 5-digit forms  
 ( "10" 4\*HEXDIG ) ; 6-digit forms  
 ; represents either "\" or a Unicode code point outside  
 ; the 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 [RFC6522] 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 standard DSN format does not permit the return of undeliverable SMTPUTF8 messages, three new media types have been defined. ([RFC5337] introduced experimental versions of these media types.)

##### 4.1. The message/global-delivery-status Media Type

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 SMTPUTF8 [RFC6531]; the "Diagnostic-Code:" field SHOULD be in i-default language [RFC2277]. 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 ASCII repertoire. These systems SHOULD up-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 utf-8-address form of a UTF-8 address type in the "Original-Recipient:" field. Third, an optional field called "Localized-Diagnostic:" is added. Each instance includes a language tag [RFC5646] 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]. Note that <text-fixed> is the same as <text> from [RFC5322], but without <obs-text>. If or when RFC 5322 is updated to disallow <obs-text>, <text-fixed> should become just <text>. Also, if or when RFC 5322 is updated to disallow control characters in <text>, <text-fixed> should become a reference to that update instead.

```
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 ]
  [ final-log-id-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.
; Updates Section 3.2.3 of RFC 3798.
;
; 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 [RFC5646].

extension-field =/ extension-field-name ":" *utf8-text
; Updates Section 7 of RFC3798

text-fixed = %d1-9 /           ; Any ASCII character except for NUL,
                        %d11 /      ; CR, and LF.
                        %d12 /      ; See note above about <text-fixed>
                        %d14-127

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

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

```

#### 4.2. The message/global Media Type

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 [RFC6532].

#### 4.3. The message/global-headers Media Type

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 SMTPUTF8 message). Unlike message/global, this body part provides no difficulties for the present infrastructure.

#### 4.4. Using These Media Types with multipart/report

Note that as far as a multipart/report [RFC6522] 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 media types mentioned above inside a multipart/report media 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 a 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 [RFC6532] relaxed a restriction from MIME [RFC2046] regarding the use of Content-Transfer-Encoding in new "message" subtypes. This specification explicitly allows the use of Content-Transfer-Encoding in message/global-headers and message/global-delivery-status. This is not believed to be problematic as these new media types are intended primarily for use by newer systems with full support for 8-bit MIME and UTF-8 headers.

#### 4.5. Additional Requirements on SMTP Servers

If an SMTP server that advertises both SMTPUTF8 and DSN needs to return an undeliverable SMTPUTF8 message, then it has two choices for encapsulating the SMTPUTF8 message when generating the corresponding multipart/report:

If the return-path SMTP server does not support SMTPUTF8, 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/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 [RFC2277].) 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 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 field, which is an address header field and thus follows the same 8-bit rules as other address header fields 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 [RFC3464]. The registration template response follows:

(a) The 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 ASCII repertoire, a specification for how they are to be encoded as graphic ASCII characters in an "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. Only the first form is 7-bit safe.

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

The mail diagnostic type registry was created by [RFC3464] and updated by [RFC5337]. This specification replaces [RFC5337]. The registration for the 'smtp' diagnostic type has been updated to reference RFC 6533 in addition to [RFC3464] and to remove the reference to [RFC5337].

When the 'smtp' diagnostic type is used in the context of a message/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 [RFC6532] 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 down-convert this media type to the text/rfc822-header media type, such conversion is discouraged as it loses information.

**Published specification:** RFC 6533

**Applications that use this media type:** SMTPUTF8 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 6533

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 6533

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

- [ASCII] American National Standards Institute (formerly United States of America Standards Institute), "USA Code for Information Interchange", ANSI X3.4-1968, 1968.
- ANSI X3.4-1968 has been replaced by newer versions with slight modifications, but the 1968 version remains definitive for the Internet.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2277] Alvestrand, H., "IETF Policy on Character Sets and Languages", BCP 18, RFC 2277, January 1998.
- [RFC3461] Moore, K., "Simple Mail Transfer Protocol (SMTP) Service Extension for Delivery Status Notifications (DSNs)", RFC 3461, 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.
- [RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, October 2008.
- [RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, October 2008.
- [RFC5646] Phillips, A. and M. Davis, "Tags for Identifying Languages", BCP 47, RFC 5646, September 2009.
- [RFC6522] Kucherawy, M., Ed., "The Multipart/Report Media Type for the Reporting of Mail System Administrative Messages", STD 73, RFC 6522, January 2012.



- [RFC6530] Klensin, J. and Y. Ko, "Overview and Framework for Internationalized Email", RFC 6530, February 2012.
- [RFC6531] Yao, J. and W. Mao, "SMTP Extension for Internationalized Email", RFC 6531, February 2012.
- [RFC6532] Yang, A., Steele, S., and N. Freed, "Internationalized Email Headers", RFC 6532, February 2012.

## 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.
- [RFC5337] Newman, C. and A. Melnikov, "Internationalized Delivery Status and Disposition Notifications", RFC 5337, September 2008.

## Appendix A. Changes since RFC 5337

Changes were made to move from Experimental to Standards Track. The most significant was the removal of an embedded alternative ASCII address within a utf-8-address, and the reflections of the ABNF changes in [RFC6531].

Fixed description of utf-8-addr-xtext and utf-8-addr-unitext.

References to Downgrade and uMailbox removed/fixed.

ABNF changes and fixed errata submitted by Alfred Hoenes.

Minor changes to MIME type references.

Other minor corrections.

## Appendix B. Acknowledgements

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

**Authors' Addresses**

Tony Hansen (editor)  
AT&T Laboratories  
200 Laurel Ave.  
Middletown, NJ 07748  
US

EMail: [tony+eaidsn@maillennium.att.com](mailto:tony+eaidsn@maillennium.att.com)

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

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

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

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