Internet Engineering Task Force (IETF)
Request for Comments: 6068
Obsoletes: 2368
Category: Standards Track
ISSN: 2070-1721

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

October 2010

The 'mailto' URI Scheme

Abstract

This document defines the format of Uniform Resource Identifiers (URIs) to identify resources that are reached using Internet mail. It adds better internationalization and compatibility with Internationalized Resource Identifiers (IRIs; RFC 3987) to the previous syntax of 'mailto' URIs (RFC 2368).

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/rfc6068.

MICROSOFT CORP.
EXHIBIT 1016

Copyright Notice

Copyright (c) 2010 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
2. Syntax of a 'mailto' URI 3
3. Semantics and Operations 7
4. Unsafe Header Fields 7
5. Encoding 8
6. Examples 9
  6.1. Basic Examples 9
  6.2. Examples of Complicated Email Addresses 10
  6.3. Examples Using UTF-8-Based Percent-Encoding 11
7. Security Considerations 12
8. IANA Considerations 14
  8.1. Update of the Registration of the 'mailto' URI Scheme . . 14
  8.2. Registration of the Body Header Field 15
9. Main Changes from RFC 2368 15
10. Acknowledgments 15
11. References 16
  11.1. Normative References 16
  11.2. Informative References 17

1. Introduction

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

This specification extends the previous scheme definition to also allow character data to be percent-encoded based on UTF-8 [STD63], which offers a better and more consistent way of dealing with non-ASCII characters for internationalization.

This specification does not address the needs of the ongoing Email Address Internationalization effort (see [RFC4952]). In particular, this specification does not include syntax for fallback addresses.

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

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

2. Syntax of a 'mailto' URI

The syntax of a 'mailto' URI is described using the ABNF of [STD68], non-terminal definitions from [RFC5322] (dot-atom-text, quoted-string), and non-terminal definitions from [STD66] (unreserved, pct-encoded):

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

<addr-spec> is a mail address as specified in [RFC5322], but excluding <comment> from [RFC5322]. However, the following changes apply:

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

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

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

4. Percent-encoding can be used in the <domain> part of an <addr-spec> in order to denote an internationalized domain name. The considerations for <reg-name> in [STD66] apply. In particular, non-ASCII characters MUST first be encoded according to UTF-8 [STD63], and then each octet of the corresponding UTF-8 sequence MUST be percent-encoded to be represented as URI characters. URI-producing applications MUST NOT use percent-encoding in domain names unless it is used to represent a UTF-8 character sequence. When the internationalized domain name is used to compose a message, the name MUST be transformed to the Internationalizing Domain Names in Applications (IDNA) encoding [RFC5891] where appropriate. URI producers SHOULD provide these domain names in the IDNA encoding, rather than percent-encoded, if they wish to maximize interoperability with legacy 'mailto' URI interpreters.

5. Percent-encoding of non-ASCII octets in the <local-part> of an <addr-spec> is reserved for the internationalization of the <local-part>. Non-ASCII characters MUST first be encoded according to UTF-8 [STD63], and then each octet of the corresponding UTF-8 sequence MUST be percent-encoded to be represented as URI characters. Any other percent-encoding of non-ASCII characters is prohibited. When a <local-part> containing non-ASCII characters will be used to compose a message, the <local-part> MUST be transformed to conform to whatever encoding may be defined in a future specification for the internationalization of email addresses.

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

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

Within ’mailto’ URIs, the characters "?", "=", and "&" are reserved, serving as delimiters. They have to be escaped (as "%3F", "%3D", and "%26", respectively) when not serving as delimiters.

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

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

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

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

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

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

is equivalent to

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

is equivalent to

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

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

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

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

Note that this specification, like any URI scheme specification, does not define syntax or meaning of a fragment identifier (see [STD66]), because these depend on the type of a retrieved representation. In the currently known usage scenarios, a ’mailto’ URI cannot be used to retrieve such representations. Therefore, fragment identifiers are

meaningless, SHOULD NOT be used on ’mailto’ URIs, and SHOULD be ignored upon resolution. The character "#" in <hfvalue>s MUST be escaped as %23.

3. Semantics and Operations

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

The operation of how any URI scheme is resolved is not mandated by the URI specifications. In current practice, resolving URIs such as those in the ’http’ URI scheme causes an immediate interaction between client software and a host running an interactive server. The ’mailto’ URI has unusual semantics because resolving such a URI does not cause an immediate interaction with a server. Instead, the client creates a message to the designated address with the various header fields set as default. The user can edit the message, send the message unedited, or choose not to send the message.

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

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

4. Unsafe Header Fields

The user agent interpreting a ’mailto’ URI SHOULD NOT create a message if any of the header fields are considered dangerous; it MAY also choose to create a message with only a subset of the header fields given in the URI. Only a limited set of header fields such as

Subject and Keywords, as well as Body, are believed to be both safe and useful in the general case. In cases where the source of a URI is well known, and/or specific header fields are limited to specific well-known values, other header fields MAY be considered safe, too.

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

5. Encoding

[STD66] requires that many characters in URIs be encoded. This affects the ’mailto’ URI scheme for some common characters that might appear in addresses, header fields, or message contents. One such character is space (" ", ASCII hex 20). Note the examples below that use "%20" for space in the message body. Also note that line breaks in the body of a message MUST be encoded with "%0D%0A". Implementations MAY add a final line break to the body of a message even if there is no trailing "%0D%0A" in the body <hfield> of the ’mailto’ URI. Line breaks in other <hfield>s SHOULD NOT be used.

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

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

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

6. Examples

6.1. Basic Examples

A URI for an ordinary individual mailing address:

<mailto:chris@example.com>

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

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

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

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

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

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

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

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

A request to subscribe to a mailing list:

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

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

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

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

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

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

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

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

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

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

As described above, the "&" (ampersand) character is reserved in HTML and has to be replaced, e.g., with "&". Thus, a URI with an internal ampersand might look like:

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

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

6.2. Examples of Complicated Email Addresses

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

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

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

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

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

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

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

6.3. Examples Using UTF-8-Based Percent-Encoding

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

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

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

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

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

```
<mailto:user@
example.org?subject=%3D%3Fiso-8859-1%3FQ%3Fcaf%3DE9%3F%3D>
```

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

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

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

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

caf=C3=A9
```

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

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

caf=E9
```

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

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

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

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

When constructing the email, the domain name label is converted to punycode. The resulting message may look as follows:

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

NATTO
```

7. Security Considerations

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

A ’mailto’ URI gives a template for a message that can be sent by mail client software. The contents of that template may be opaque or difficult to read by the user at the time of specifying the URI, as well as being hidden in the user interface (for example, a link on an HTML Web page might display something other than the content of the corresponding ’mailto’ URI that would be used when clicked). Thus, a mail client SHOULD NOT send a message based on a ’mailto’ URI without first disclosing and showing to the user the full message that will be sent (including all header fields that were specified by the ’mailto’ URI), fully decoded, and asking the user for approval to send the message as electronic mail. The mail client SHOULD also make it clear that the user is about to send an electronic mail message, since the user may not be aware that this is the result of a ’mailto’ URI. Users are strongly encouraged to ensure that the ’mailto’ URI presented to them matches the address included in the "To:" line of the email message.

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

Examples of problems with sending unapproved mail include:

mail that breaks laws upon delivery, such as making illegal threats;

mail that identifies the sender as someone interested in breaking laws;

mail that identifies the sender to an unwanted third party;

mail that causes a financial charge to be incurred by the sender;

mail that causes an action on the recipient machine that causes damage that might be attributed to the sender.

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

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

Programs manipulating ’mailto’ URIs have to take great care to not inadvertently double-escape or double-unescape ’mailto’ URIs, and to make sure that escaping and unescaping conventions relating to URIs and relating to mail addresses are applied in the right order.

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

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

8. IANA Considerations

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

This document changes the definition of the 'mailto' URI scheme; the registry of URI schemes has been updated to refer to this document rather than its predecessor, [RFC2368]. The registration template is as follows:

URI scheme name:
'mailto'

Status:
permanent

URI scheme syntax:
See the syntax section of RFC 6068.

URI scheme semantics:
See the semantics section of RFC 6068.

Encoding considerations:
See the syntax and encoding sections of RFC 6068.

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

Interoperability considerations:
Interoperability for 'mailto' URIs with UTF-8-based percent-encoding might be somewhat lower than interoperability for 'mailto' URIs with US-ASCII only.

Security considerations:
See the security considerations section of RFC 6068.

Contact:
IETF

Author/Change controller:
IETF

References:
RFC 6068

8.2. Registration of the Body Header Field

IANA has registered the Body header field in the Message Header Fields Registry ([RFC3864]) as follows:

Header field name:
Body

Applicable protocol:
None. This registration is made to assure that this header field name is not used at all, in order to not create any problems for 'mailto' URIs.

Status:
reserved

Author/Change controller:
IETF

Specification document(s):
RFC 6068

Related information:
none

9. Main Changes from RFC 2368

The main changes from RFC 2368 are as follows:

- o Changed syntax from RFC 2822 <mailbox> to [RFC5322] <addr-spec>.
- o Allowed UTF-8-based percent-encoding for domain names and in <hfvalue>.
- o Nailed down percent-encoding in <local-part> to be based on UTF-8, reserved for if and when there is a specification for the internationalization of email addresses.
- o Removed prohibition against "Bcc:" header fields, but added a warning about their visibility and harvesting for spam.
- o Added clarifications for escaping.

10. Acknowledgments

This document was derived from [RFC2368]; the acknowledgments from that specification still apply. In addition, we thank Paul Hoffman for his work on [RFC2368].

Valuable input on this document was received from (in no particular order): Alexey Melnikov, Paul Hoffman, Charles Lindsey, Tim Kindberg, Frank Ellermann, Etan Wexler, Michael Haardt, Michael Anthony Puls II, Eliot Lear, Dave Crocker, Dan Harkins, Nevil Brownlee, John Klensin, Alfred Hoenes, Ned Freed, Sean Turner, Peter Saint-Andre, Adrian Farrel, Avshalom Houri, Robert Sparks, and many others.

11. References

11.1. Normative References

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

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

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

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

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

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

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

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

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

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

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

11.2. Informative References

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

[RFC2368] Hoffman, P., Masinter, L., and J. Zawinski, "The mailto URL scheme", RFC 2368, July 1998.

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

Authors' Addresses

Martin Duerst (Note: Please write "Duerst" with u-umlaut wherever possible, for example as "Dürst" in XML and HTML.)
Aoyama Gakuin University
5-10-1 Fuchinobe
Sagamihara, Kanagawa 229-8558
Japan

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

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

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

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

EMail: jwz@jwz.org