INTERNET ARCHIVE
archive.org

AFFIDAVIT OF NATHANIEL E FRANK-WHITE

1. I am a Records Request Processor at the Internet Archive. I make this declaration of my own personal knowledge.

2. The Internet Archive is a website that provides access to a digital library of Internet sites and other cultural artifacts in digital form. Like a paper library, we provide free access to researchers, historians, scholars, and the general public. The Internet Archive has partnered with and receives support from various institutions, including the Library of Congress.

3. The Internet Archive has created a service known as the Wayback Machine. The Wayback Machine makes it possible to browse more than 450 billion pages stored in the Internet Archive's web archive. Visitors to the Wayback Machine can search archives by URL (i.e., a website address). If archived records for a URL are available, the visitor will be presented with a display of available dates. The visitor may select one of those dates, and begin browsing an archived version of the Web. Links on archived files in the Wayback Machine point to other archived files (whether HTML pages or other file types), if any are found for the URL indicated by a given link. For instance, the Wayback Machine is designed such that when a visitor clicks on a hyperlink on an archived page that points to another URL, the visitor will be served the archived file found for the hyperlink's URL with the closest available date to the initial file containing the hyperlink.

4. The archived data made viewable and browsable by the Wayback Machine is obtained by use of web archiving software that automatically stores copies of files available via the Internet, each file preserved as it existed at a particular point in time.

5. The Internet Archive assigns a URL on its site to the archived files in the format http://web.archive.org/web/[Year in yyyy][Month in mm][Day in dd][Time code in hh:mm:ss]/[Archived URL] aka an "extended URL". Thus, the extended URL http://web.archive.org/web/19970126045828/http://www.archive.org/ would be the URL for the record of the Internet Archive home page HTML file (http://www.archive.org/) archived on January 26, 1997 at 4:58 a.m. and 28 seconds (1997/01/26 at 04:58:28). The date indicated by an extended URL applies to a preserved instance of a file for a given URL, but not necessarily to any other files linked therein. Thus, in the case of a page constituted by a primary HTML file and other separate files (e.g., files with images, audio, multimedia, design elements, or other embedded content) linked within that primary HTML file, the primary HTML file and the other files will each have their own respective extended URLs and may not have been archived on the same dates.

6. Attached hereto as Exhibit A are true and accurate copies of browser screenshots of the Internet Archive's records of the archived files for the URLs and the dates specified in the attached coversheet of each printout.

MICROSOFT CORP.
EXHIBIT 1014

Document Id: BD92A3B0-6D5B-11EE-AAA5-6BFB22FBE7C8
OnlineNotary.net

INTERNET ARCHIVE
archive.org

7. Attached hereto as Exhibit B are true and accurate copies of the Internet Archive's records of the archived files for the URLs and the dates specified in the attached coversheet of each file.

8. I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct.

DATE: 10/17/2023

Nathaniel Frank-White
Nathaniel E Frank-White

Please see attached
All Purpose
Jurat form
for additional
Notary Events

Document Id: BD92A3B0-6D5B-11EE-AAA5-6BFB22FBE7C8
OnlineNotary.net

JURAT ATTACHMENT

A notary public or other officer completing this certificate verifies only the identity of the individual who signed the document to which this certificate is attached, and not the truthfulness, accuracy, or validity of that document.

STATE OF Texas }

COUNTY OF Tarrant }

The foregoing instrument was subscribed and sworn before me this date of 10/17/2023, by Nathaniel Frank-White

This notarial act was an online notarization.



(Notary Seal)

Notary's Signature _____

Registration No.: 133645929

Commission Expiration Date: March 15, 2026

Document Id: BD92A3B0-6D5B-11EE-AAA5-6BFB22FBE7C8
OnlineNotary.net

EXHIBIT A

https://web.archive.org/web/20110120223245/https://www.ietf.org/rfc/rfc2368.txt

INTERNET ARCHIVE WayBackMachine https://www.ietf.org/rfc/rfc2368.txt Go AUG JAN APR 345 captures 7 Sep 2000 - 29 Sep 2023 2010 20 2011 2012 About this capture

Network Working Group P. Hoffman
Request for Comments: 2368 Internet Mail Consortium
Updates: 1738, 1808 L. Masinter
Category: Standards Track Xerox Corporation
J. Zawinski
Netscape Communications
July 1998

The mailto URL scheme

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1998). All Rights Reserved.

Abstract

This document defines the format of Uniform Resource Locators (URL) for designating electronic mail addresses. It is one of a suite of documents which replace RFC 1738, 'Uniform Resource Locators', and RFC 1808, 'Relative Uniform Resource Locators'. The syntax of 'mailto' URLs from RFC 1738 is extended to allow creation of more RFC 822 messages by allowing the URL to express additional header and body fields.

1. Introduction

The mailto URL scheme is used to designate the Internet mailing address of an individual or service. In its simplest form, a mailto URL contains an Internet mail address.

For greater functionality, because interaction with some resources may require message headers or message bodies to be specified as well as the mail address, the mailto URL scheme is extended to allow setting mail header fields and the message body.

2. Syntax of a mailto URL

Following the syntax conventions of RFC 1738 [RFC1738], a "mailto" URL has the form:


```
mailtoURL  =  "mailto:" [ to ] [ headers ]
to         =  #mailbox
headers    =  "?" header *( "&" header )
header     =  hname "=" hvalue
hname      =  *urlc
hvalue     =  *urlc
```

"#mailbox" is as specified in RFC 822 [RFC822]. This means that it consists of zero or more comma-separated mail addresses, possibly including "phrase" and "comment" components. Note that all URL reserved characters in "to" must be encoded: in particular, parentheses, commas, and the percent sign ("%"), which commonly occur in the "mailbox" syntax.

"hname" and "hvalue" are encodings of an RFC 822 header name and value, respectively. As with "to", all URL reserved characters must be encoded.

The special hname "body" indicates that the associated hvalue is the body of the message. The "body" hname should contain the content for the first text/plain body part of the message. The mailto URL is primarily intended for generation of short text messages that are actually the content of automatic processing (such as "subscribe" messages for mailing lists), not general MIME bodies.

Within mailto URLs, the characters "?", "=", "&" are reserved.

Because the "&" (ampersand) character is reserved in HTML, any mailto URL which contains an ampersand must be spelled differently in HTML than in other contexts. A mailto URL which appears in an HTML document must use "&" instead of "&".

Also note that it is legal to specify both "to" and an "hname" whose value is "to". That is,

mailto:addr1%2C%20addr2

is equivalent to

mailto:?to=addr1%2C%20addr2

is equivalent to

mailto:addr1?to=addr2

8-bit characters in mailto URLs are forbidden. MIME encoded words (as defined in [RFC2047]) are permitted in header values, but not for any part of a "body" hname.

3. Semantics and operations

A mailto URL designates an "internet resource", which is the mailbox specified in the address. When additional headers 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 URL is not intended as a way of retrieving such objects automatically.

In current practice, resolving URLs such as those in the "http" scheme causes an immediate interaction between client software and a host running an interactive server. The "mailto" URL has unusual semantics because resolving such a URL does not cause an immediate interaction. 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 this message unedited, or choose not to send the message. The operation of how any URL scheme is resolved is not mandated by the URL specifications.

4. Unsafe headers

The user agent interpreting a mailto URL SHOULD choose not to create a message if any of the headers are considered dangerous; it may also choose to create a message with only a subset of the headers given in the URL. Only the Subject, Keywords, and Body headers are believed to be both safe and useful.

The creator of a mailto URL cannot expect the resolver of a URL to understand more than the "subject" and "body" headers. Clients that resolve mailto URLs into mail messages should be able to correctly create RFC 822-compliant mail messages using the "subject" and "body" headers.

5. Encoding

RFC 1738 requires that many characters in URLs be encoded. This affects the mailto scheme for some common characters that might appear in addresses, headers or message contents. One such character is space (" ", ASCII hex 20). Note the examples above 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".

People creating mailto URLs must be careful to encode any reserved characters that are used in the URLs so that properly-written URL interpreters can read them. Also, client software that reads URLs must be careful to decode strings before creating the mail message so

that the mail messages appear in a form that the recipient will understand. These strings should be decoded before showing the user the message.

The mailto URL scheme is limited in that it does not provide for substitution of variables. Thus, a message body that must include a user's email address can not be encoded using the mailto URL. This limitation also prevents mailto URLs that are signed with public keys and other such variable information.

6. Examples

URLs for an ordinary individual mailing address:

<mailto:chris@example.com>

A URL for a mail response system that requires the name of the file 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 URL could have 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 your mailto URL is when browsing archives of messages. Each browsed message might contain a mailto URL like:

<mailto:foobar@example.com?In-Reply-To=%3c3469A91.D10AF4C@example.com>

A request to subscribe to a mailing list:

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

A URL for a single user which includes a CC of another user:

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

Another way of expressing the same thing:

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

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

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

According to RFC 822, the characters "?", "&", and even "%" may occur in addr-specs. The fact that they are reserved characters in this URL scheme is not a problem: those characters may appear in mailto URLs, they just may not appear in unencoded form. The standard URL encoding mechanisms ("%" followed by a two-digit hex number) must be used in certain cases.

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

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

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

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

As described above, the "&" (ampersand) character is reserved in HTML and must be replacded with "&". Thus, a complex URL that has internal ampersands might look like:

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

7. Security Considerations

The mailto 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 encoded, they can also be read at any of those sites.

A mailto URL 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 URL. Thus, a mail client should never send a message based on a mailto URL without first showing the user the full message that will be sent (including all headers that were specified by the mailto URL), 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 URL.

A mail client should never send anything without complete disclosure to the user of what is will be sent; it should disclose not only the message destination, but also any headers. Unrecognized headers, or headers with values inconsistent with those the mail client would normally send should be especially suspect. MIME headers (MIME-Version, Content-*) are most likely inappropriate, as are those relating to routing (From, Bcc, Apparently-To, etc.).

relating to routing (From, Bcc, Apparently-To, etc.)

Note that some headers are inherently unsafe to include in a message generated from a URL. For example, headers such as "From:", "Bcc:", and so on, should never be interpreted from a URL. In general, the fewer headers interpreted from the URL, 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 on 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 URLs should ensure that the SMTP "From" address is set and correct.

8. IANA Considerations

This document changes the definition of the mailto: URI scheme; any registry of URI schemes should refer to this document rather than its predecessor, RFC 1738.

9. References

[RFC822] Crocker, D., "Standard for the Format of ARPA Internet Text Messages", STD 11, RFC 822, August 1982.

[RFC1738] Berners-Lee, T., Masinter, L., and M. McCahill, Editors, "Uniform Resource Locators (URL)", RFC 1738, December 1994.

[RFC1808] Fielding, R., "Relative Uniform Resource Locators", RFC 1808, June 1995.

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

A. Change from RFC 1738

RFC 1738 defined only a simple 'mailto' with no headers, just an addr-spec (not a full mailbox.) However, required usage and implementation has led to the development of an extended syntax that included more header fields.

B. Acknowledgments

This document was derived from RFC 1738 and RFC 1808 [RFC1808]; the acknowledgments from those specifications still applies.

The following people contributed to this memo or had and discussed similar ideas for mailto.

Harald Alvestrand
Bryan Costales
Steve Dorner
Al Gilman
Mark Joseph
Laurence Lundblade
Keith Moore
Jacob Palme
Michael Patton

C. Author Contact Information

Paul E. Hoffman
Internet Mail Consortium
127 Segre Place
Santa Cruz, CA 95060 USA

EMail: phoffman@imc.org

Larry Masinter
Xerox Corporation
3333 Coyote Hill Road
Palo Alto, CA 94304 USA

EMail: masinter@parc.xerox.com

Jamie Zawinski
Netscape Communications Corp.
501 East Middlefield Road
Mountain View, CA 94043 USA

EMail: jwz@netscape.com

D. Full Copyright Statement

Copyright (C) The Internet Society (1998). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

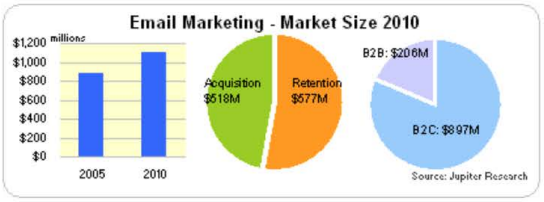
This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

https://web.archive.org/web/20070128213417/http:/www.metrics2.com/blog/2006/11/07/email_marketing_stats_facts_and_metrics_metrics_20.html

INTERNET ARCHIVE WayBackMachine http://www.metrics2.com/blog/2006/11/07/email_marketing_stats_facts_and_metrics_metrics_20.html Go DEC JAN MAR 28 2006 2007 2008 About this capture
68 captures 13 Dec 2006 - 4 Mar 2023

Email Marketing Stats, Facts and Metrics - Metrics 2.0 Quick Pack

Direct Marketing Association's latest research highlights that Email marketing in the US will account for approximately 71,000 jobs this year, growing at 8-10% annually historically and projected into the future as well. Additionally, the ROI for email marketing is $57.25 for every dollar spent. The ROI of all non-email online marketing is $22.52, less than half. Separately, Jupiter Research expects email marketing spend to hit $1.1 billion by 2010.



Silverpop CEO Bill Nussey points out that email provides more than twice the ROI of other Internet marketing channels, but gets only 2.5% of the investment.

Return Path has just received a patent for its Email Change of Address (ECOA) service. The service has been used by more than 400 companies, and more than 31 million consumers have registered with the service. On average, 15,000 people register with Return Path's ECOA service daily.

To help size up this key marketing channel, here is a succinct compilation of key email marketing stats, facts, and metrics:

Email Marketing Market Size

E-mail marketing spending will reach $1.1 billion in 2010 from $885 million in 2005, growing at a compounded annual rate of 4.5%, according to a report from Jupiter Research. The highest growth will be from transactional e-mail, increasing 37% over the next five years to $7 million.

Spending on retention e-mail accounts for the greatest share of the e-mail market at $577 million by 2010. Acquisition e-mail spending, including newsletter sponsorship, append and co-registration, will grow 4% annually from 2005 to 2010 to $518 million.

B2C Email marketing spend is estimated to grow 5% annually to $897 million by 2010, while B2B spending will grow by 2.4% annually to $206 million in 2010.

Meanwhile, the volume of spam messages per consumer will drop 13% annually to 1,640 pieces of spam from 3,253 pieces, Jupiter said. By 2010, the cost of incorrectly blocked e-mail will drop to $92 million from a high of $107 million this year.

Email Marketing Importance:

According to a recent WebTrends® report, retailers ranked email marketing as the most important demand-generation activity for holiday success, with Search Engine Marketing (SEM) and Search Engine Optimization (SEO) ranking second and third. The biggest increases in marketing spending for the holidays will be in email marketing (52%), SEM (46%) and SEO (38%). The biggest decreases in spending will be in online banner ads (17%), print advertising (16%) and broadcast advertising (14%). WebTrends estimates that 80% of retailers use regular customer emails to build customer relationships.

A recent study of 270 retail sites (pdf) by Email Data Source finds that, in 2005, only 29% of the retailers surveyed had any type of email marketing program in place while in 2006, that figure had jumped to 70%, an increase of 141%. In 2005, only 31% of retailers who had an email program in place actually sent any email at all, including welcome letters. In 2006, 67% sent at least one email within a 30 day period, showing an increase of 116% in a single year.

In 2005, only 2% of those retail sites with an email program in place required a double opt-in. In 2006, that figured jumped to 5%, a 150% increase, which is much better, but according to best practice recommendations still leaves lots of room for improvement.

	2005	2006	% Change
Retailers with Email Marketing Programs	29%	70%	141%
Retailers who actually sent Emails	31%	67%	116%
Retailers requiring double opt-in	2%	5%	150%

Source: Email Data Source

Overall stats of the retailers who had an email program in place, according to Email Data Source:

- 33.33% sent no email within the first 30 days.
- 21.29% sent follow-up email but no welcome message.
- 15.66% sent a welcome message but never sent additional email.
- 29.72% sent a welcome message and followed it up with regular communication.
- 4.82% sent a double opt-in email, confirming the initial sign-up.
- 92.41% had a sign-up form on the front page of their web site.
- 70.04% required additional information such as a name or zip code.

By The Numbers

Global IPO Activity by Value, 2006	$253 billion
U.S. Government IT Spending, 2007	$150 billion
U.S. Retaile E-Commerce Spending, 2006	$102 billion
Pharmaceutical industry R&D spend, 2004	$40 billion
Global Electronic Equipment Sales, 2007	$1.47 trillion
Global Semiconductor Revenues, 2007	$285 billion
Semiconductor Capital Equipment Spend, 2007	$56.6 billion

by Metrics2 Get it here

All the information on this site is for informational purposes only, not for investment purposes or advice.
Copyright 2006 Metrics 2.0 - All rights reserved.
About | Contact

Email Marketing Delivery and Effectiveness:

According to Harte-Hanks Postfuture Index™ for January-June 2006, among all sectors for all purposes of e-mail combined,

- the average delivered rate stands at 91.2%
- average open rate of 78.8% of those e-mail delivered
- click-through rate of 18.4% of all e-mail delivered
- and an opt-out rate of 0.4% of all e-mail delivered

Based on 44.4 million email messages, Bronto's trends in email marketing benchmark for the 3-month period ending November 4, 2006, reports the following average email delivery stats:

- Delivered (95.6%)
- Opened(28.9%)
- Clickthoughs (5.4%)

A recent Return Path study shows that 83% of email delivery problems (blocking and filtering) are the result of reputation issues, not content. The study, based on 550 email campaigns, shows that sender reputation causes 77% of delivery failures, while an additional 6% are triggered by the reputation of domains included in the email content and and only 17% being related to factors such as words, fonts, images, or spelling.

Jupiter Research found that e-mail delivery rates have stabilized at an average 88% and are expected to surpass 90% over the next few years.

Another recent Jupiter survey finds 54% of respondents saying they clicked on a link in an e-mail because the product or service interested them.

Making simple changes to creative elements of email messages can boost click rates by 50% or more, according to new research conducted by Silverpop. Email subject line, ratio of images to text, and the email layout style are the key creative elements evaluated in this study.

- BtoB emails with the company or brand name in the subject line experienced an average open rate of 32%, compared to just 20% for messages without branded subject lines.
- While all-text BtoC messages generated a 4.7% click rate on average, those that were mostly image-oriented achieved a 7.1% average click rate.

More Email Industry Statistics from Ferris Research

- Estimated number of non-spam email messages sent worldwide each day, 2006: 25 billion
- % of email messages sent daily that are spam messages: > 60%
- Typical # Internet emails sent and received by a business user: 600 per week
- Cost of a user deleting a spam message: $0.04
- Cost of a user retrieving a bona fide message erroneously deleted as spam ("false positive"): $3.50

Ferris Research's estimate of the world's largest email systems, measured in number of active mailboxes (accessed at least once over the last 30 days), as of 10/05:

The World's Largest Email Systems.

Organization	Seats (Millions)
Sun *	150
OpenWave *	100
Hotmail	100
Microsoft	100
Critical Path *	60
AOL **	80
Yahoo	40
IBM Notes/Domino	25
Google Gmail	20
Novell GroupWise	10

** indicates a vendor that specializes in selling to service providers*
*** Includes free seats*
Source: Ferris Research

Business Email Users (millions)

	2005	2006	2007	2008	2009	2010
North America	205.7	216.2	227.0	238.0	249.2	260.6
Europe	162.6	179.8	196.5	212.8	228.6	244.1
Other Americas	179.1	191.9	204.7	217.4	230.2	243.0
Africa	16.0	19.5	23.0	26.6	30.1	33.7
Asia (incl. Mid-East)	182.8	198.3	213.6	229.0	244.3	259.6
Oceania	8.7	9.1	9.5	9.9	10.4	10.8
Total	549.1	598.5	647.3	695.6	743.6	791.2

Source: Ferris Research, The Email Security Market, 2005-2010
Figures are in millions of users, rounded to the nearest 100,000.

BOOKMARK

November 7, 2006 2:45 PM | Permalink

Previous: « Online Video Ad Dollars to Jump 82.2%; Contribute Only 2.6% of Online Ad Biz

Next : » U.S. Consumer Credit Fell $1.2B, the Biggest Drop Since April 1992

INTERNET ARCHIVE WayBackMachine http://www.mailchimp.com/articles/email_marketing_guide Go SEP OCT NOV 218 captures 21 Feb 2009 - 23 May 2023 2008 02 2009 2010 About this capture

Email Marketing Articles and Guides

Resources
Articles & Guides
Merge Tag Cheatsheet
ChimpCharts
Template Language
HTML Email Templates
Find an Expert
MailChimp Jungle
MailChimp API
MailChimp Plugins

← Back to all articles

Download Now

Designing, Coding And Delivering HTML Email

This free, 65-page guide covers all the basics for delivering bulletproof HTML emails. It's perfect if you're new to email marketing, and you want a crash course (on everything). We'll cover how HTML emails work, so you can have a basic understanding of the fundamentals. Then, we'll dive into technical stuff (design and coding HTML emails) then best practices, deliverability (dealing with spam filters), and how to track your email marketing performance.

What's Inside The Guide:

Everything you ever wanted to know about sending email newsletters and email marketing campaigns. We'll give you answers to these common questions:

- How do HTML emails work, and why do they keep breaking when I try to send them through my own email program (Microsoft Outlook, Apple Mail, Entourage, etc)?
- How do I insert or attach pictures in my HTML email newsletters?
- What tools and software do I need to create HTML emails?
- What are some basic guidelines for designing HTML emails?
- What makes a good HTML email design?
- How do spam filters work, and how can I avoid getting my emails trapped?
- Which email programs will work with my HTML emails, and which won't?
- How can I test and troubleshoot my email designs?
- How do I track opens and clicks in my email campaigns?
- What do all the spam laws mean, and can I get in trouble with email marketing?

About MailChimp

MailChimp's a DIY email marketing service that's chock full of useful, powerful email marketing features. Over 175,000 companies of all shapes and sizes use MailChimp to send their email campaigns.
Try MailChimp free today

Other Email Marketing Resources:

If you're new to email marketing, you might also like these free resources.

- Download free HTML Email Templates
- How Spam Filters Work
- Improving Your Email Open Rates

MailChimp Blog

Be sure to check out the MailChimp Blog for email marketing news, tips, best practices, monkeys, and more.

Popular Articles & Guides

- How to Code HTML Emails
- Email Marketing Subject Line Comparison
- Best Practices in Writing Email Subject Lines
- Email Marketing Guide

- HTML Email Designer

- Free Image Hosting
- Email Autoresponders

- Use MailChimp for Free!
- Get Started

Free HTML Templates →

Templates tested in all major email clients, ready for customizing.

Compare Us →

Check out how we compare against other email marketing providers.

View Pricing Plans →

MailChimp offers flexible plans for every budget.

Forever Free Plan →

Manage up to 500 subscribers, send up to 3,000 emails/month for free.

Follow us on Twitter · Subscribe to our Newsletter · MailChimp Jungle · MailChimp Labs

Copyright © 2009 MailChimp. All rights reserved · Contact Us · Partner With Us · Abuse Desk · Privacy Policy · Copyright Policy · Terms of Use

INTERNET ARCHIVE WayBackMachine
http://www.unsubcentral.com/index.php
Go
139 captures
2 Apr 2004 - 4 Jun 2011
NOV JAN FEB
2010 04 2011 2012
About this capture

UNSUBCENTRAL
beyond compliance

RSS Feed | 1.800.589.0445

Solutions CAN-SPAM Customers Company Client Login

Save Time, Save Money and Be Compliant

The Leader in Email Compliance and Suppression List Management

Learn more >>

1 2 3



Be Compliant



OVERVIEW: UnsubCentral provides email compliance and suppression list management solutions through a secure and automated platform that simplifies suppression list updates and distribution across all parties while minimizing risks, protecting brands, often lowering costs and helping ensure compliance. Learn more >

Follow Us | Resources | Case Studies



Facebook
Blog
Twitter
LinkedIn

UnsubNews

UnsubCentral and PreferenceCentral Honored in Online Trust Alliance Annual Leadership Awards

Advertising Week
Join us discuss the future of privacy in the age of trust

Online Trust Alliance Recognizes Excellence in Online Trust Awards

Suppression List | Opt-Out | CAN-SPAM | MD5 Distribution

Suppression List

Suppression list management is the method in which an advertiser or network collects and distributes opt-out requests from and to their media partners in the interest of compliance. It is the responsibility of networks, advertisers and publishers to effectively manage and distribute suppression lists, assuring that all opt-out requests are honored within the ten day time frame that the law allows. This means that you must continually update your suppression file, both internal opt-outs and those generated by third parties, and share them with any media partners on a regular basis to ensure that recent opt-outs are provided before your media partners proceed with a mailing. Failure to honor a request within the ten day time frame can result in a judgment against you for "Failure to Honor" one of the tenants of CAN-SPAM. UnsubCentral stays well versed in CAN-SPAM and Privacy legislation, for more information visit the Federal Trade Commission

HOME | PRIVACY POLICY | CONTACT US © 2009 UNSUBCENTRAL. All rights reserved. TRUSTe CERTIFIED PRIVACY

https://web.archive.org/web/20111007034343mp_/http://www.ietf.org/rfc/rfc6068.txt

INTERNET ARCHIVE WayBackMachine http://www.ietf.org/rfc/rfc6068.txt Go SEP OCT AUG 88 captures 7 Oct 2011 - 17 Sep 2023 07 2010 2011 2013 About this capture

Internet Engineering Task Force (IETF) M. Duerst
Request for Comments: 6068 Aoyama Gakuin University
Obsoletes: 2368 L. Masinter
Category: Standards Track Adobe Systems Incorporated
ISSN: 2070-1721 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.

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

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

<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%5C%22it's%5C%20ugly%5C%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:

- Changed syntax from RFC 2822 <mailbox> to [RFC5322] <addr-spec>.
- Allowed UTF-8-based percent-encoding for domain names and in <hfvalue>.
- 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.
- Removed prohibition against "Bcc:" header fields, but added a warning about their visibility and harvesting for spam.
- 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

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

https://web.archive.org/web/20100504200607mp_/http://www.ietf.org/rfc/rfc2369.txt

INTERNET ARCHIVE WayBackMachine http://www.ietf.org/rfc/rfc2369.txt Go APR MAY JUN 215 captures 11 Apr 2000 - 4 Aug 2023 2009 04 2010 2011 About this capture

Network Working Group G. Neufeld
Request for Comments: 2369 Nisto
Category: Standards Track J. Baer
SkyWeyr Technologies
July 1998

The Use of URLs as Meta-Syntax for Core Mail List Commands and their Transport through Message Header Fields

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1998). All Rights Reserved.

Abstract

The mailing list command specification header fields are a set of structured fields to be added to email messages sent by email distribution lists. Each field typically contains a URL (usually mailto [RFC2368]) locating the relevant information or performing the command directly. The three core header fields described in this document are List-Help, List-Subscribe, and List-Unsubscribe.

There are three other header fields described here which, although not as widely applicable, will have utility for a sufficient number of mailing lists to justify their formalization here. These are List-Post, List-Owner and List-Archive.

By including these header fields, list servers can make it possible for mail clients to provide automated tools for users to perform list functions. This could take the form of a menu item, push button, or other user interface element. The intent is to simplify the user experience, providing a common interface to the often cryptic and varied mailing list manager commands.

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

1. Introduction

This is a proposal for additional header fields to be added to email messages sent by email distribution lists. The content of each new field is typically a URL - usually mailto [RFC2368] - which locates the relevant information or performs the command directly. MTAs generating the header fields SHOULD usually include a mailto based command, in addition to any other protocols used, in order to support users who do not have access to non-mail-based protocols.

Implementing these fields will be optional. Significant functionality and convenience can be gained by including them, however. Many list managers, especially as the proposal first gains acceptance, MAY choose to implement only one or two of the fields. The List-Help field is the most useful individual field since it provides an access point to detailed user support information, and accommodates almost all existing list managers command sets. The List-Subscribe and List-Unsubscribe fields are also very useful, but cannot describe some list manager syntaxes at this time (those which require variable substitution). See appendix A.5 for an explanation.

The description of command syntax provided by the fields can be used by mail client applications to provide simplified and consistent user access to email distribution list functions. This could take the form of menu items, push buttons, or other user interface elements. The intent is to simplify the user experience, providing a common interface to the often cryptic and varied mailing list manager commands.

Consideration has been given to avoiding the creation of too many fields, while at the same time avoiding the overloading of individual fields and keeping the syntax clear and simple.

The use of these fields does not remove the requirement to support the -Request command address for mailing lists [RFC2142].

2. The Command Syntax

The list header fields are subject to the encoding and character restrictions for mail headers as described in [RFC822]. Additionally, the URL content is further restricted to the set of URL safe characters [RFC1738].

The contents of the list header fields mostly consist of angle-bracket ('<', '>') enclosed URLs, with internal whitespace being ignored. MTAs MUST NOT insert whitespace within the brackets, but client applications should treat any whitespace, that might be inserted by poorly behaved MTAs, as characters to ignore.

A list of multiple, alternate, URLs MAY be specified by a comma-separated list of angle-bracket enclosed URLs. The URLs have order of preference from left to right. The client application should use the left most protocol that it supports, or knows how to access by a separate application. By this mechanism, protocols like http may be specified while still providing the basic mailto support for those clients who do not have access to non-mail protocols. The client should only use one of the available URLs for a command, using another only if the first one used failed.

The use of URLs allows for the use of the syntax with existing URL supporting applications. As the standard for URLs is extended, the list header fields will gain the benefit of those extensions. Additionally, the use of URLs provides access to multiple transport protocols (such as ftp and http) although it is expected that the "mailto" protocol [RFC2368] will be the focus of most use of the list header fields. Use of non-mailto protocols should be considered in light of those users who do not have access to the specified mechanism (those who only have email - with no web access).

Command syntaxes requiring variable fields to be set by the client (such as including the user's email address within a command) are not supported by this implementation. However, systems using such syntaxes SHOULD still take advantage of the List-Help field to provide the user with detailed instructions as needed or - perhaps more usefully - provide access to some form of structured command interface such as an HTML-based form.

The additional complications of supporting variable fields within the command syntax was determined to be too difficult to support by this protocol and would compromise the likelihood of implementation by software authors.

To allow for future extension, client applications MUST follow the following guidelines for handling the contents of the header fields described in this document:

1) Except where noted for specific fields, if the content of the field (following any leading whitespace, including comments) begins with any character other than the opening angle bracket '<', the field SHOULD be ignored.

2) Any characters following an angle bracket enclosed URL SHOULD be ignored, unless a comma is the first non-whitespace/comment character after the closing angle bracket.

3) If a sub-item (comma-separated item) within the field is not an angle-bracket enclosed URL, the remainder of the field (the current, and all subsequent, sub-items) SHOULD be ignored.

3. The List Header Fields

This document presents header fields which will provide the command syntax description for the 'core' and key secondary functions of most email distribution lists. The fields implemented on a given list SHOULD be included on all messages distributed by the list (including command responses to individual users), and on other messages where the message clearly applies to one distinct list. There MUST be no more than one of each field present in any given message.

These fields MUST only be generated by mailing lists, not end users.

3.1. List-Help

The List-Help field is the most important of the header fields described in this document. It would be acceptable for a list manager to include only this field, since by definition it SHOULD direct the user to complete instructions for all other commands. Typically, the URL specified would request the help file, perhaps incorporating an HTML form for list commands, for the list, and alternatively provide access to an instructive website.

Examples:

```
List-Help: <mailto:list@host.com?subject=help> (List Instructions)
List-Help: <mailto:list-manager@host.com?body=info>
List-Help: <mailto:list-info@host.com> (Info about the list)
List-Help: <http://www.host.com/list/>, <mailto:list-info@host.com>
List-Help: <ftp://ftp.host.com/list.txt> (FTP),
    <mailto:list@host.com?subject=help>
```

3.2. List-Unsubscribe

The List-Unsubscribe field describes the command (preferably using mail) to directly unsubscribe the user (removing them from the list).

Examples:

```
List-Unsubscribe: <mailto:list@host.com?subject=unsubscribe>
List-Unsubscribe: (Use this command to get off the list)
    <mailto:list-manager@host.com?body=unsubscribe%20list>
List-Unsubscribe: <mailto:list-off@host.com>
```



```
List-Unsubscribe: <http://www.host.com/list.cgi?cmd=unsub&lst=list>,
    <mailto:list-request@host.com?subject=unsubscribe>
```

3.3. List-Subscribe

The List-Subscribe field describes the command (preferably using mail) to directly subscribe the user (request addition to the list).

Examples:

```
List-Subscribe: <mailto:list@host.com?subject=subscribe>
List-Subscribe: <mailto:list-request@host.com?subject=subscribe>
List-Subscribe: (Use this command to join the list)
    <mailto:list-manager@host.com?body=subscribe%20list>
List-Subscribe: <mailto:list-on@host.com>
List-Subscribe: <http://www.host.com/list.cgi?cmd=sub&lst=list>,
    <mailto:list-manager@host.com?body=subscribe%20list>
```

3.4. List-Post

The List-Post field describes the method for posting to the list. This is typically the address of the list, but MAY be a moderator, or potentially some other form of submission. For the special case of a list that does not allow posting (e.g., an announcements list), the List-Post field may contain the special value "NO".

Examples:

```
List-Post: <mailto:list@host.com>
List-Post: <mailto:moderator@host.com> (Postings are Moderated)
List-Post: <mailto:moderator@host.com?subject=list%20posting>
List-Post: NO (posting not allowed on this list)
```

3.5. List-Owner

The List-Owner field identifies the path to contact a human administrator for the list. The URL MAY contain the address of a administrator for the list, the mail system administrator, or any other person who can handle user contact for the list. There is no need to specify List-Owner if it is the same person as the mail system administrator (postmaster).

Examples:

```
List-Owner: <mailto:listmom@host.com> (Contact Person for Help)
List-Owner: <mailto:grant@foo.bar> (Grant Neufeld)
List-Owner: <mailto:josh@foo.bar?Subject=list>
```


3.6. List-Archive

The List-Archive field describes how to access archives for the list.

Examples:

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

List-Archive: <mailto:archive@host.com?subject=index%20list>
List-Archive: <ftp://ftp.host.com/pub/list/archive/>
List-Archive: <http://www.host.com/list/archive/> (Web Archive)

4. Supporting Nested Lists

A list that is a sublist for another list in a nested mailing list hierarchy will need to modify some of the List- header fields, while leaving others as the parent list set them.

Sublists SHOULD remove the parent list's List-Help, List-Subscribe, List-Unsubscribe and List-Owner fields, and SHOULD insert their own versions of those fields.

If the sublist provides its own archive, it SHOULD replace the List-Archive with its own. Otherwise, it MUST leave the List-Archive field untouched.

Dependant on how postings to the list are handled, the sublist MAY replace the List-Post field. The appropriateness of whether to replace List-Post is left to the determination of the individual list managers. If the intention is that postings should be distributed to all members of the primary list, List-Post should not be changed by a sublist in such a way that postings will be distributed only to members of the sublist.

5. Security Considerations

There are very few new security concerns generated with this proposal. Message headers are an existing standard, designed to easily accommodate new types. There may be concern with multiple fields being inserted or headers being forged, but these are problems inherent in Internet email, not specific to the protocol described in this document. Further, the implications are relatively harmless.

Mail list processors should not allow any user-originated list header fields to pass through to their lists, lest they confuse the user and have the potential to create security problems.

On the client side, there may be some concern with posts or commands being sent in error. It is required that the user have a chance to confirm any action before it is executed. In the case of mailto, it

may be appropriate to create the correctly formatted message without sending it, allowing the user to see exactly what is happening and giving the user the opportunity to approve or discard the message before it is sent.

All security considerations for the use of URLs [RFC1738] apply equally to this protocol. Mail client applications should not support list header field URLs which could compromise the security of the user's system. This includes the "file://" URL type which could potentially be used to trigger the execution of a local application on some user systems.

6. Acknowledgements

The numerous participants of the List-Header [5], ListMom-Talk [6], List-Managers and MIDA-Mail mailing lists contributed much to the formation and structure of this document.

Keith Moore <moore@cs.utk.edu> and Christopher Allen <ChristopherA@consensus.com> provided guidance on the standards process.

A. Background Discussion

This proposal arose from discussions started on the ListMom-Talk Discussion List [6]. When the discussion reached a sufficient level, a separate list was formed for discussing this proposal, the List Headers Mail List [5] for deeper discussion. We have included summaries of key issues raised, in order to show some of the alternatives examined and reasons for our decisions.

A.1. Multiple header fields vs. a single header field

Use of a single header field for transporting command meta-syntax was rejected for a number of reasons.

Such a field would require the creation of a new meta-syntax in order to describe the list commands (as opposed to the use of the widely deployed URL syntax which was chosen for this implementation). Every additional layer of complexity and newness reduces the likelihood of actual implementation because it will require additional work to support. Also, by using the existing URL syntax, we can profit from the end users' knowledge of that syntax and ability to use it even if their client applications do not support the list header fields.

Restricting the transport of meta-syntax to the use of a single header field also introduces complications with header field size limitations. Most individual commands can easily be described in a single line, but describing a multitude of commands can take up many lines in the field and runs a greater risk of being modified by an existing server on route.

The client implementation is also easier with multiple fields, since each command can be supported and implemented individually, completely independent of the others. Thus, some list managers or mail clients can choose to implement a subset of the fields based on the specific needs of their individual lists.

Finally, the format described in this document is simple and well recognized, which reduces the chances of errors in implementation and parsing.

A.2. URLs vs. parameter lists

URLs are already an established syntax which is flexible, well-defined, and in wide spread use. As its definition matures and expands, the abilities of the list fields will grow as well, without requiring modification of this proposal. URLs are well prepared to handle future protocols and developments, and can easily describe the different existing access protocols such as mailto, http and ftp.

Many clients already have functionality for recognizing, parsing, and evaluating URLs, either internally or by passing the request to a helper application. This makes implementation easier and more realistic. As an example, this existing support for URL parsing allowed us to add prototype list header functionality to existing mail clients (Eudora and Emailer for the Macintosh) without modifying their source code.

A.3. Why not just create a standard command language?

A standard command language, supported by all email list services, would go a long way to reducing the problems of list access that currently plague existing services. It would reduce the amount of learning required by end users and allow for a number of common support tools to be developed.

However, such standardization does pose problems in the areas of multi-lingual support and the custom needs of individual mailing lists. The development of such a standard is also expected to be met with a slow adoption rate by software developers and list service providers.

These points do not preclude the development of such a standard (in fact, it would suggest that we should start sooner rather than later), but we do need a solution that can be widely supported by the current list services.

We can support most existing list manager command syntaxes without a standard command language. By using URLs, we allow alternate access methods a standard command language probably wouldn't enable, such as web based control.

Finally, client support for a standard command language is not at all clear or necessarily simple to implement. The variety and large number of commands existing today would require complicated user interfaces which could be confusing and difficult to implement. By restricting this proposal to the core functions, the client

implementation is much simpler, which significantly increases the likelihood of implementation (as evidenced by the support already announced by a number of client and server application authors).

A.4. Internationalization

Multilingual support is up to the URL standard. If URLs support it, then the List- header fields support it. This is another advantage of using URLs as the building blocks for the list header fields.

A.5. Variable Substitution

Variables would allow the List- header fields to accommodate nearly every existing list manager. However, it would immeasurably increase the complexity of the entire proposal, and possibly involve redefining the URL standard, or force us to use something more complicated (and hence more difficult to implement) than URLs to describe the command syntax.

Parameters would either have to be mandatory (i.e. the user agent doesn't submit the message if it doesn't know what text to substitute) or you need a way to say "if you know this parameter, add its text here; otherwise, do this" where "this" is either: (a) substitute a constant string, or (b) fail.

The reason you would want a facility like this is because some list server applications insist on having certain parameters like users' names, which the user agent might or might not know. e.g. listserv insists on having a first name and a last name if you supply either one.

Which could lead to something like the UNIX shell syntax, where ${foo-bar} means substitute the value of parameter "foo" if "foo" is defined, else substitute the string "bar". Perhaps $foo would mean "substitute the value of parameter foo if it is defined, else substitute the empty string"

This all seems far too complicated for the gains involved, especially since the use of variables can often be avoided.

The use of variables in the command syntaxes of list services appears to be lessening and does not, in any case, apply to all commands. While the unsubscribe and subscribe command header fields may not be usable by those systems which require the use of variables, the help field will still provide end users with a consistent point of access through which they can get support for their use of the list.

A.6. Why not use a specialized MIME part instead of header fields?

MIME parts were considered, but because most mail clients currently either don't support MIME or are not equipped to handle such specialized parts - such an implementation would result in problems for end users. It is also not as easy for many list servers to implement MIME as it is to implement new header fields.

However, we are looking at the design of a MIME part to more fully describe list command syntax, as well as trying to find ways to get it supported by the applicable software.

A.7. Why include a Subscribe command?

Subscribe and Unsubscribe are the key commands needed by almost every list. Other commands, such as digest mode, are not as widely supported.

Additionally, users who have unsubscribed (before going on vacation, or for whatever other reason) may want to resubscribe to a list. Or, a message may be forwarded/bounced from a subscriber to a non-subscriber. Or, the user may change addresses and want to subscribe from their new address. Having the List-Subscribe field available could certainly help in all these cases.

A.8. The Dangers of Header Bloat

At what point are there just too many header fields? It really varies on a list by list basis. On some lists, the majority of users will never be aware of a field unless the client software provides some alternative user interface to it (akin to the Reply-To field). On others, the users will often see the header fields of messages and would be able to recognize the function of the URLs contained within.

The flexibility afforded by the protocol described in this document (in that the header fields may be individually implemented as deemed appropriate) provides list administrators with sufficient 'room to

appropriate) provides list administrators with sufficient 'room to maneuver' to meet their individual needs.

B. Client Implementation

B.1. Guidelines

For 'mailto' URL based commands, mail client applications may choose to provide specialized feedback (such as presenting a dialog or alert), instead of the actual command email message, asking for command confirmation from the user. The feedback should identify the message destination and command within a more descriptive explanation. For example:

> "Do you want to send the unsubscription command 'unsubscribe somelist' to 'somelist-request@some.host.com'? Sending the command will result in your removal from the associated list."

If the user has multiple email addresses supported by the mail client, the client application should prompt the user for which address to use when subscribing or performing some other action where the address to use cannot be specifically determined. When unsubscribing or such, the address that is subscribed should be used, unless that is not known by the application and cannot be determined from the message headers.

B.2. Implementation Options

The following implementation possibilities are suggested here to give some idea as to why these new header fields will be useful, and how they could be supported.

In most cases, it may be helpful to disable the interface for the commands when not applicable to the currently selected message.

B.2.1. Key combinations and command lines

On text based systems which utilize command lines or key combinations, each field could be implemented as a separate command. Thus one combination would subscribe the user, another would unsubscribe, a third request help, etc. The commands would only be available on messages containing the list header fields.

B.2.2. Menu items

On graphical systems which have menus, these commands could take the form of a menu or sub-menu of items. For example, a "Lists" menu might appear when viewing messages containing the header fields, with items named "Subscribe", "Unsubscribe", "Get Help", "Post Message to

List", "Contact List Owner" and "Access List Archive". This menu could be disabled when not applicable to the current message or disappear entirely.

B.2.3. Push Buttons and Pallettes

On graphical window systems, buttons could be placed in the window of the message, a toolbar, or in a floating pallette of their own. Each button could correspond to a command, with names "Subscribe", "Unsubscribe", "Get Help", "Post to List", "List Owner" and "Archive". These buttons or pallettes could be disabled when not applicable to the current message or disappear entirely.

B.2.4 Feedback to the User

   If using a dialog interface (or other feedback element) the client application MUST include an option for the user to review (and possibly modify) the message before it is sent. The application may also find it useful to provide a link to more detailed context-sensitive assistance about mail list access in general.

References

   [RFC822] Crocker, D., "Standard for the Format of ARPA Internet Text Messages", STD 11, RFC 822, August 1982.

   [RFC1738] Berners-Lee, T., Masinter, L., and M. McCahill, "Uniform Resource Locators (URL)" RFC 1738, December 1994.

   [RFC2142] Crocker, D., "Mailbox Names for Common Services, Roles and Functions", RFC 2142, May 1997.

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

   [5] "List-Header" Mail list. list-header@list.nisto.com
       <URL:http://www.nisto.com/listspec/mail/>
       <URL:http://www.nisto.com/listspec/>

   [6] "ListMom-Talk" Mail list. listmom-talk@skyweyr.com
       <URL:http://cgi.skyweyr.com/ListMom.Home>

Editors' Addresses

   Joshua D. Baer
   Box 273
   4902 Forbes Avenue
   Pittsburgh, PA 15213-3799
   USA

   EMail: josh@skyweyr.com

   Grant Neufeld
   Calgary, Alberta
   Canada

   EMail: grant@acm.org
   Web: http://www.nisto.com/

Full Copyright Statement

Copyright (C) The Internet Society (1998). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS 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.

https://web.archive.org/web/20110711170032/http://www.ceas.cc/2008/Participation-2008.html

INTERNET ARCHIVE WayBackMachine http://www.ceas.cc/2008/Participation-2008.html Go JUN JUL 11 2011 OCT 2010 2015 About this capture
5 captures 11 Jul 2011 - 26 May 2022

CALL FOR PARTICIPATION

CEAS 2008 -- Fifth Conference on Email and Anti-Spam

Aug 21 and 22, 2008 (Thursday, Friday)

Mountain View, California

The organizers of the Conference on Email and Anti-Spam invite you to participate in its fourth annual event. This forum brings together academic and industrial researchers to present new work in all aspects of email and messaging, including uses and abuses such as spam.

A special event associated with CEAS 2008 is the Spam Challenge

For update on

* Accepted papers
* Invited Speakers
* Conference schedule
* Spam Challenge
* Conference facility
* Hotel information

please visit:

http://www.ceas.cc

On-line Early Registration:

https://www.acteva.com/booking.cfm?bevaid=157864

Until July 20:	Student	--	$100
	Non-student	--	$250
After July 20:	Student	--	$150
	Non-student	--	$300
On-site:	Student	--	$200
	Non-student	--	$370

Hotel:

http://www.ceas.cc/2008/hotel2008
Before July 21st, if booked by phone: $109.99

General Conference Chair

Aleksander Kolcz (Microsoft Live Labs)

Program Co-Chairs

Andrej Bratko (Jozef Stefan Institute)
Barry Leiba (IBM Research)
Tobias Scheffer (Max Planck Institute for Computer Science)

Compitition Co-Chairs

Andrej Bratko (Jozef Stefan Institute)
Richard Segal (IBM Research)

President

Gordon V. Cormack (University of Waterloo)

Questions?

Contact: mailto information at ceas.cc
(received by all the chairs)

https://web.archive.org/web/20091215072504fw_/http://www.ceas.cc/2008/program_2008.html

	Thursday, August 21	Friday, August 22
8:00-9:00	Breakfast Service	
8:45-9:00	**Opening**	
9:00-10:00	**Keynote by Lois Greisman** Federal Trade Commission	**Keynote by Bradley Taylor** Fighting Spam: Gmail's Story
10:00-10:20	Coffee Break	
10:20-12:00	**Filtering Techniques** Improved Phishing Detection using Model-Based Features *Andre Bergholz, Jeong-Ho Chang, Gerhard Paass, Frank Reichartz, Siehyun Strobel* Filtering Email Spam in the Presence of Noisy User Feedback *D. Sculley, Gordon Cormack* Exploiting Transport-Level Characteristics of Spam *Robert Beverly, Karen Solins* Personalized Spam Filtering for Gray Mail *Ming-Wei Chang, Scott Yih, Robert McCann*	**Email Usage and Applications** Reducing E-Discovery Cost by Filtering Included Emails *Tsuen-Wan Ngan* Toward a stochastic speech act model of email behavior *John Mildinhall, Jan Noyes* Activity-centred Search in Email *Einat Minkov, Ramnath Balasubramanyan, William Cohen* **Short Paper:** Do Zebras get more Spam than Aardvarks? *Richard Clayton*
12:00-1:40	**Lunch Break (catered)**	
1:40-3:20	**Trusted Friends** Improving Email Trustworthiness through Social-Group Key Authentication *Vivek Pathak, Danfeng Yao, Liviu Iftode* Tracking Email Reputation for Authenticated Sender Identities *Gautam Singaraju, Jeffrey Moss, Brent ByungHoon Kang* Social Honeypots: Making Friends With A Spammer Near You *Steve Webb, James Caverlee, Calton Pu* On Free Speech and Civil Discourse: Filtering Abuse in Blog Comments *D. Sculley*	**Filtering and Evaluation** Breaking out of the Browser to Defend Against Phishing Attacks *Diana Smetters, Paul Stewart* Resolving FP-TP Conflict in Digest-Based Collaborative Spam Detection by Use of Negative Selection Algorithm *Slavisa Sarafijanovic, Sabrina Perez, Jean-Yves Le Boudec* Global-scale Anti-spam Testing in Your Own Back Yard *Margaret Nielsen, Dane Bertram, Sampson Pun, John Aycock, Nathan Friess* A Mail Client Plugin for Privacy-Preserving Spam Filter Evaluation *Mona Mojdeh, Gordon Cormack*
3:20-3:50	Coffee Break	
3:50-5:40	**Sending Spam** A Survey of Modern Spam Tools *Henry Stern* A Campaign-based Characterization of Spamming Strategies *Pedro Calais, Douglas Pires, Dorgival Guedes, Wagner Meira Jr., Cristine Hoepers, Klaus Steding-Jessen* Detecting Known and New Salting Tricks in Unwanted Emails *Andre Bergholz, Gerhard Paass, Frank Reichartz,&#65533; Siehyun Strobel, Marie-Francine Moens, Brian Witten&#65533;* **Short Papers** Improving Image Spam Filtering Using Image Text Features *Giorgio Fumera, Fabio Roli, Battista Biggio, Ignazio Pillai&#65533;* Analysis of Spectral Parameters of Audio Signals for the Identification of Spam Over IP Telephony *Christoph P&#65533;rschmann, Heiko Knospe*	**Black- and Whitelisting** The Effectiveness of Whitelisting: a User-Study *David Erickson* Empirical research on IP blacklisting *Christian Dietrich, Christian Rossow* **4:40 Business Meeting**
6:00-7:00	**Bar Service**	
from 7:00	**Conference Banquet and Poster Session**	

https://web.archive.org/web/20151004021251/http://www.ceas.cc/2008/volunteer.html

INTERNET ARCHIVE
WayBackMachine
http://www.ceas.cc/2008/volunteer.html
Go
SEP
OCT 04 2015
AUG
2014
2020
4 captures
4 Oct 2015 - 26 May 2022
About this capture

CALL FOR STUDENT VOLUNTEERS

CEAS 2008 -- Fifth Conference on Email and Anti-Spam
Aug. 21 and 22, 2008 (Thursday, Friday)

Mountain View, California
www.ceas.cc

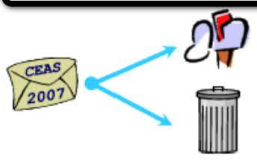
Students are invited to contribute to the CEAS 2008 conference as volunteers. A variety of tasks will be undertaken by student volunteers, such as conference registration, setting up presentation equipments, etc. These tasks are important to the success of the conference, and at the same time provide student volunteers valuable opportunities to interact with other CEAS participants and fellow volunteers, many of whom are working in the Email and Anti-Spam research area.

As a benefit, student volunteers are offered free registration to the CEAS 2008 conference. You must be present from Aug. 21 to Aug. 22, and commit to approximately 8 hours of work. Prior experience is not required. Preference will be given to students who work on a similar or related research area.

If you are interested in becoming a student volunteer, please send an email to: alek at ir dot iit dot edu. Please include a resume or CV in your email.

https://web.archive.org/web/20081005074846/http:/www.ceas.cc/

INTERNET ARCHIVE WayBackMachine http://www.ceas.cc/ Go SEP OCT 05 JAN 2007 2008 2010 About this capture
596 captures 17 Jan 2004 - 11 Oct 2023



Fifth Conference on Email and Anti-Spam
CEAS 2008

Aug 21-22, 2008

Microsoft Research Silicon Valley, Mountain View, California

Authors: Copyright Form
2008 Program Committee
Conference Program
Call for Student Volunteers
Spam Filtering Challenge
Registration
Hotel and Venue
Call for Participation
CEAS 2007
CEAS 2006
CEAS 2005
CEAS 2004

Distinguished Sponsors

AOL Google Microsoft proofpoint TREND MICRO

Spam Challenge Sponsors

IBM PROJECT HONEY POT

Invited Speakers

Lois Greisman, FTC

https://web.archive.org/web/20110102213012/http://www.constantcontact.com/display_image.jsp?image_url=screenshots/EM_lists_safeunsubscribe.gif

INTERNET ARCHIVE
WayBackMachine
http://www.constantcontact.com/display_image.jsp?image_url=screenshots/EM_lists_safeunsubscribe.gif
Go
NOV JAN JUN
139 captures
24 May 2010 - 14 Apr 2013
2010 02 2012
2011
About this capture

Close

Please Confirm Your Unsubscribe

Before unsubscribing, is this you?

c*******@gmail.com**

If this is not your email address, there is no need to unsubscribe. You have not been added to any mailing lists.

Are you sure you wish to stop ALL emails from **Sunset Travel Online** sent to your email address?

Yes, unsubscribe me from all mailings | Cancel

Sunset Travel Online uses **SafeUnsubscribe®** which guarantees the permanent removal of your email address from its mailing lists.

INTERNET ARCHIVE WayBackMachine
http://returnpath.net/blog/whitepapers.php
Go
76 captures
12 Nov 2007 - 30 Jun 2023
MAR JUN SEP
2009 11 2010 2011
About this capture

Return Path Research

Return Path research and publications provide a wealth of knowledge regarding best practices, client experiences, and industry research studies.

- Best Practice Guides
- Research Studies
- Client Success Stories

Stay Informed

Check out our IN The Know blog and monthly newsletter to stay current on Return Path's cutting edge research.

- Subscribe to our IN The Know Newsletter
- Read our IN The Know blog

Resources & Publications

Best Practice Guides

- The Super Six: Best Practices That Will Make You More Successful with Email Marketing
- Your Reputation Holds the Key to Deliverability
- Tactics to Combat and Manage ISP Image Blocking
- Personal Whitelisting: How Subscribers Can Contribute to Deliverability

Research Studies

- 2010 Email Delivery Imperatives
- Top Email Trends in 2009
- The Global Email Deliverability Benchmark Report, 2H 2009
- Emailing in the Dark: What European Email Marketers Don't Understand about Deliverability
- Telling Stories, Building Relationships: How Non-Profits Can Create More Engaging Email Marketing Programs
- The Email Deliverability Benchmark Report, 1H 2009
- Increasing Revenues by Optimizing Emailing Practices with Online Buyers
- Keeping the Subscriber Experience Positive After "Unsubscribe Me"
- Q2 2008 Reputation Benchmark Report
- Fourth Annual Holiday Email Consumer Survey
- Creating Great Subscriber Experiences: Are Marketers Relationship Worthy?

Client Success Stories

Looking for client success stories? We've moved them to a new section.

Privacy | Copyright 2010 Return Path, Inc.

https://web.archive.org/web/20090131154140/http://www.lashback.com:80/products/UnsubSafe.aspx

INTERNET ARCHIVE WayBackMachine
http://www.lashback.com/products/UnsubSafe.aspx
Go
22 captures
20 May 2007 - 22 Aug 200
DEC JAN MAR
2007 31 2009 2010
About this capture

Overview
UnsubMonitor
CAN-SPAM Monitor
BrandAlert
UnsubSafe
Get a Demo
Feature Comparison Chart

ISP and Receiver UnsubSafe

The First and Only

Simply filtering junk email only hides the problem. To truly receive relief from the ever increasing flood of junk email sent to your customers through your network, LashBack brings you its revolutionary new UnsubSafe service. Available to Internet Service Providers and anti-spam firms free of charge, UnsubSafe immediately reduces the amount of junk email sent to you and your customers each day.

In order to determine which email marketers can be trusted with your unsubscribe request, the UnsubSafe service automatically searches though its massive database of email marketer's reputations and extracts an UnsubScore. Based on the score that is returned by UnsubScore, LashBack knows when it's safe to unsubscribe and when requests should be safely kept away from un-trusted email marketers.

More than One Way to UnsubSafe

Option 1

If you currently allow your customers to report spam, you can simply forward a copy of these messages to LashBack via secure SMTP. Alternatively, if your network already filters out spam messages sent to your customers, you can simply forward a copy of filtered message to LashBack. In either case, LashBack will parse through those messages to extract the email marketer's identity and unsubscribe reputation, or UnsubScore. If the email marketer has a good reputation, LashBack will automatically and safely unsubscribe your customer from future mailings. If the email marketer has a bad unsubscribe reputation, the unsubscribe attempt is suppressed.

UnsubSafe SDK Documentation
Is your business ready for UnsubSafe? Contact Us.

Option 2

If your organization doesn't already have a network-level spam filtering service deployed, or you would like to extend spam reporting all the way to your customer, LashBack can integrate a customizable reporting toolbar that is directly installed into the customers email client (Outlook and Outlook Express).

Deploying the UnsubSafe toolbar solution allows your organization the opportunity to increase the perceived value of your service by enabling your customers to see and experience the value of UnsubSafe first hand. Additionally, customization of the toolbar with your organization's logo further deepens the bond between you and your customers.

"89.7% of respondents agreed that they would prefer that their ISP or email service provider include an unsubscribe option that would safely remove them from email lists"
Bigfoot Interactive - Spam Research Survey

Expand your Spam Intelligence
Optionally, when you deploy the UnsubSafe toolbar on your network, you can have a copy of all messages reported on your network sent back to you for review. This data can be used by your to gauge the impact and success of your current filtering solutions as well as additional intelligence to maximize your networks catch rate and filtering performance.

Reporting
Real-time Unsubscribe Performance reports are made available so you can see the value and impact LashBack is having on your organization's junk email problem. At a glance, you'll be able to review network level statistics and unsubscribe requests as well as additional data you can use to drill into any specific spam report and view each email marketers' UnsubScore and the messages they sent your customers.

Contact LashBack Sales at 800.434.1994

To learn more about Option 2, contact us.

© 2003-2007 LashBack, LLC.
555 Washington Ave. Suite 400, St. Louis, MO 63101
(800) 434-1994

https://web.archive.org/web/20090131072744/http://www.lashback.com:80/reputation/DataStory.aspx

INTERNET ARCHIVE WayBackMachine
http://www.lashback.com/reputation/DataStory.aspx
Go
OCT JAN 31 2007 2009 FEB 2010
18 captures
30 Apr 2007 - 2 Feb 2009
About this capture

lashback™

800.434.1994
request a demo | newsletter signup

products solutions news support contact us LOGIN

reputation
trusted or untrusted?

- Reputation
- Data Story
- Reputation Toolbox
- Unsubscribe Best Practices
- Blacklist Resources

Feature Comparison Chart

Data Story: Return to Sender

Where Does LashBack Data Come From? How are you "always watching"?

The Power of the LashBack Compliance FeedBack Loop is derived from real-world consumer data. LashBack receives data from many valuable sources. The most valuable data is from the free LashBack consumer toolbar for Outlook and Outlook Express (link to New consumer toolbar download page.) The toolbar provides an automatic check of the reputation of the owner of the unsubscribe mechanism, which enables consumers to unsubscribe safely from unwanted email.

LashBack also works with a number of ISPs, anti-spam organizations and ESP's who feed LashBack monitors a data stream of email for analysis. LashBack is analyzing over 150,000 pieces of actual consumer emails on a daily basis. The data stream is real-world and uniquely consumer driven. The LashBack relationship to consumer inboxes is the most vital relationship our client's leverage.

The very best reputation management services companies and receivers utilize LashBack's unique data as part of the reputation score with which they each rate senders.

This process is also called a "feedback loop" which all email marketing experts agree email senders need to be a part of to understand the actual performance of their email marketing programs.

© 2003-2007 LashBack, LLC.
555 Washington Ave. Suite 400, St. Louis, MO 63101
(800) 434-1994

https://web.archive.org/web/20090202091204/http://www.lashback.com:80/reputation/BlacklistResources.aspx

INTERNET ARCHIVE WayBackMachine
http://www.lashback.com/reputation/BlacklistResources.aspx
Go DEC FEB AUG
16 captures
20 Jul 2007 - 30 Aug 2009
2007 02 2009 2010
About this capture

Reputation
Data Story
Reputation Toolbox
Unsubscribe Best Practices
Blacklist Resources

Feature Comparison Chart

Inc. 2007 Best of Technology
CAN-SPAM Compliance Monitor
for making sure your outbound email works

Unsubscribe Blacklist (UBL) Resources

What is the UBL?

The Unsubscribe Blacklist (UBL) is a real-time blacklist of IP addresses which are sending email to names harvested from suppression files.

Statistics

New IPs in the last 48 hours	0
IPs Removed in the last 48 hours	19,370
Delist Requests in the last 48 hours	55
Total IPs Listed	894,766

Quick Links

How to perform DNS lookups
UBL via XML
UBL via TXT
UBL in RBLDNSD Format
RSYNC Access
Research IPs listed on UBL
Blacklist Delist Procedure
Research URL Reputations
How to Build a Good Reputation

About the UBL

Utilizing the world's largest unsubscribe intelligence database, LashBack's Unsubscribe Blacklist (UBL) empowers your organization to identify and block the very worst unsubscribe abusers; those that send email to addresses which have been harvested from suppression lists.

LashBack has been monitoring sender unsubscribe compliance since 2003. During this period, LashBack has compiled sender unsubscribe compliance failure in four primary categories:

1. Which senders fail to provide a working unsubscribe mechanism
2. Which senders do not honor unsubscribe requests within 10 business days
3. Which senders allow consumers unsubscribe request to be abused (i.e., shared with other parties to send more spam)
4. Which senders are taking suppression lists from other organizations and sending email to these lists

LashBack's UBL only lists the IP addresses of senders who are using email addresses harvested from suppression lists (point 4). If you would like additional "premium" reputation data contact a LashBack representative directly.

This free service is intended to provide all receivers and reputation providers a baseline of data to help block the sender's with the worst reputations. All IP addresses listed on in our service are being continually monitored and the list is updated hourly.

Web Lookup

To perform ad-hoc lookups or to delist an IP for the first time, please visit the UBL Lookup and delist page.

XML Download

If you wish to download an XML copy of the blacklist, this file is available at the XML UBL download link. This XML file is updated hourly with the latest

blacklist data.

Text Download

If you wish to download a text copy of the blacklist, this file is available at the text UBL download link. This text file is updated hourly with the latest blacklist data.

RBLDSND Format Download

If you wish to download a copy of the blacklist in RBLDNSD format, this file is available here. This file is updated hourly with the latest blacklist data.

RSYNC Access

All downloadable files are available via RSYNC at:
rsync://rsync.unsubscore.com/LBBL/blacklist.xml (XML)
rsync://rsync.unsubscore.com/LBBL/blacklist.txt (TEXT)
rsync://rsync.unsubscore.com/LBBL/ubl_rbldns.txt (RBLDNSD)

DNS Lookups

This service is a list of hostnames in the ubl.unsubscore.com zone. These hostnames may appear to be real IP Addresses. However, they are hosts in the ubl.unsubscore.com DNS zone.

In order to check whether an IP is listed in the UBL, you can perform a DNS query using the IP address of the sender. However, you must reverse the IP address in order to perform the query. For example, if you wish to check whether 192.168.1.100 is listed in the UBL, you would perform a DNS lookup on 100.1.168.192.ubl.unsubscore.com. If one of the DNS return codes listed below is returned, then the IP is listed in the UBL.

UBL Return Codes

Currently the LashBack UBL supports the following return codes. Please note that LashBack will likely continue to add additional return codes over time.

127.0.0.2 - This return code indicates that the IP address has sent to email LashBack unsubscribe probes. These senders are taking email addresses from unsubscribe requests and spamming them.

Zone Transfers

Currently the UBL allows anonymous zone transfers of ubl.unsubscore.com.

Mail Server Configuration

The LashBack UBL can be used for any mail server supporting connection level blacklists. The blacklist server should be set to ubl.unsubscore.com.

Delisting Requests

LashBack will actively address any requests to be delisted. Before sending us a delist request, please visit the UBL Lookup and delist page, where you may delist the IP at the click of a button.

Data Use Restrictions

The UBL database is provided free for both commercial and non-commercial use.

DISCLAIMER OF WARRANTIES AND LIABILITY

All data is provided for your personal information only, and not intended for legal purposes. LashBack LLC is not liable for any errors in the data, or for any actions taken in reliance thereon.

LASHBACK LLC MAKES NO WARRANTY, EXPRESSED OR IMPLIED, AS TO THE RESULTS TO BE OBTAINED BY ANY USER OR OTHERS FROM THE USE OF THE INFORMATION PROVIDED HEREUNDER. LASHBACK MAKES NO EXPRESS OR IMPLIED WARRANTIES OF FITNESS OF THE INFORMATION FOR A PARTICULAR PURPOSE OR USE.

The information contained in LashBack databases has been developed by LashBack and are considered by LashBack to be reliable. However, the accuracy and completeness thereof are not guaranteed and LashBack and its data suppliers shall have no liability for errors or omissions regardless of the cause or source of such error or omission.

LASHBACK LLC DOES NOT HAVE ANY LIABILITY TO A USER OR TO

OTHERS FOR ANY LOST PROFITS OR INDIRECT, SPECIAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES, EVEN IF ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH DAMAGES.

LashBack may change or discontinue the databases, or their availability to you, at any time without notice.

© 2003-2007 LashBack, LLC.
555 Washington Ave. Suite 400, St. Louis, MO 63101
(800) 434-1994

https://web.archive.org/web/20090131072826/http://lashback.com:80/reputation/UnsubscribeBestPractices.aspx

INTERNET ARCHIVE WayBackMachine
http://lashback.com/reputation/UnsubscribeBestPractices.aspx
Go OCT JAN 31 JUN
17 captures
29 Apr 2007 - 23 Aug 2013
2007 2009 2013
About this capture

reputation
trusted or untrusted?

Reputation
Data Story
Reputation Toolbox
Unsubscribe Best Practices
Blacklist Resources
Feature Comparison Chart

Unsubscribe Best Practices

Confirming Unsubscribe Requests

LashBack does not support nor recommend a double-opt out process. Consumers wishing to unsubscribe from receiving email from you should not have to receive more email from you in order to stop receiving email from you. However, email notices of a successful opt-out request are recommended.

Unsubscribe Links vs. Unsubscribe Mechanisms

It is important that the distinction between unsubscribe links and unsubscribe mechanisms be made. An unsubscribe mechanism is the final email address or web page that an unsubscribe request is submitted to, ignoring any parameters. An unsubscribe link is the link the user selected to reach that mechanism. For email-based unsubscribes, the unsubscribe mechanism and unsubscribe link are considered the same. However, in web cases, the link clicked in an email (the unsubscribe link) and the ultimate page the unsubscribe is posted to (the unsubscribe mechanism) is generally different. The following example clarifies the distinction between and unsubscribe link and an unsubscribe mechanism:

Step 1 - Consumer clicks the following link in an email to unsubscribe:
http://www.domain.com/unsubscribe.htm

Step 2 - Consumer is automatically redirected to another page to fill in an email address and submit a form to unsubscribe:
http://www.domain.com/unsubscribeform.htm

Step 3 - Consumer submits the unsubscribe form which is posted to:
http://www.domain.com/processunsubscribe.cgi

Step 4 - Consumer is taken to a confirmation page indicating a successful unsubscribe:
http://www.domain.com/unsubsuccess.htm

In the above scenario, http://www.domain.com/unsubscribe.htm is considered the unsubscribe link and http://www.domain.com/processunsubscribe.cgi is considered the unsubscribe mechanism.

Ensuring LashBack Can Find your Unsubscribe Link

There are currently no standard labeling practices in the world of email unsubscribe. In order to find your unsubscribe link, LashBack uses multiple approaches, some of which involve human identification. The easier it is for LashBack to locate your unsubscribe link, the more likely you will have a good reputation with LashBack. Some general recommendations to ensure LashBack is able to find your unsubscribe link are as follows:

- Use a List-Unsubscribe header as described in RFC 2369
- In the case of email-based unsubscribe links included in the body of the email message, include the word 'unsubscribe' in the email address itself (e.g., unsubscribe-newsletter1@domain.com)
- In the case of web-based unsubscribe links included in the body of the email message, include the word 'unsubscribe' in the unsubscribe link (e.g., http://www.domain.com/unsubscribe-newsletter1.cgi or http://www.domain.com/unsubscribe/newsletter1.cgi)

Ensuring LashBack Can Use your Unsubscribe Link

Equally important to finding your unsubscribe link, LashBack must be able to easily and reliability navigate and use your unsubscribe link. Some general recommendations are listed below:

- Use an email-based unsubscribe link and make sure the actual recipient's email address (and only this email address) is specified in the TO: header. If both of these conditions are met, LashBack will always be able to reliably use your unsubscribe link.
- Web-based unsubscribe links should be hosted on a single web page (i.e., users should not have to navigate multiple pages to unsubscribe).
- Web-based unsubscribe links should pass in all required unsubscribe parameters or only require the consumer's email address to unsubscribe.
- Web-based unsubscribe links should require users to click a button to confirm the unsubscribe request. Automatically unsubscribing user's by simply clicking the unsubscribe link is not recommended.
- Web-based unsubscribe links should provide the consumer confirmation of a successful unsubscribe which includes the words 'unsubscribe' and 'successful'.
- Web-based unsubscribe links should not use or require active scripting to function (JavaScript or VBScript).

Changing Unsubscribe Mechanisms

Every time a sender changes unsubscribe mechanisms, LashBack checks to see if the sender has used the mechanism in the past. If the sender has not used the mechanism in the past, all UnsubSafe processing is suspended while LashBack tests the safety of the new mechanism. This is to ensure consumers are protected from potential misuse of their unsubscribe request. Senders wanting to ensure UnsubSafe is enabled should avoid switching unsubscribe mechanisms often. Senders can change their unsubscribe link as often as they like without suspending UnsubSafe so long as the ultimate unsubscribe mechanism remains constant.

Testing Your Unsubscribe Link with LashBack

To assist senders in checking to ensure LashBack is able to find and use their unsubscribe link, LashBack provides a simple method to do so. Simply send a message with your advertisement or newsletter to unsubcheck@unsubsafe.com. Within 5 minutes of receiving the message, an email message will be returned to the sender of the email (the FROM: address) with results indicating whether LashBack was able to a) find your unsubscribe link and b) use your unsubscribe link. If you are having problems getting your unsubscribe link to work with LashBack, LashBack may be able to assist you with making changes.

Expectations for Segmenting Unsubscribes

While the CAN SPAM Act dictates that senders must provide consumers an option to opt out, it is sometimes unclear about what unsubscribe options should be provided to consumers and how unsubscribe requests should apply within. LashBack's unsubscribe segmentation expectations are designed to match consumer expectations. However, LashBack does not warranty that the segmentation expectations here will protect senders from legal liability under the CAN SPAM Act. Senders and advertisers should always consult legal counsel in making these policy decisions. When in doubt, Senders should err on the side of caution and treat requests globally.

Opt In Communications
In general, unsubscribe segmentation should match opt-in segmentation. For example, if a consumer explicitly signs up for a newsletter with an organization, then a consumer would expect that an unsubscribe request would eliminate this future correspondence. In cases where a consumer has opted in for general communication from your organization but this was not specific to a specific type of communication, then opt out requests should be treated globally in the organization.

Acquisition Emails
All emails sent advertising products or services to customers that do not have a preexisting relationship with the actual seller/provider of the product of service. In cases of acquisition email, all unsubscribes should be applied globally within the organization. If there are two parties involved in the transmission of the email – a sender and an advertiser – both parties should provide a global unsubscribe link accessible from the email sent. Segmenting unsubscribes by offer, brand, domain or other method is not accepted to qualify for UnsubSafe.

Mirror Unsubscribe Mechanisms with Unsubscribe Segmentation
LashBack uses several techniques to determine when a sender or advertiser is not honoring unsubscribe requests. One of the primary techniques LashBack uses to identify repeat attempts for a consumer to unsubscribe using the same unsubscribe mechanism after 10 business days (failure to honor). Senders and advertisers should be aware of this, since using the same unsubscribe mechanism for multiple lists may cause false positives in their unsubscribe reputation. LashBack CSRs review all unsubscribe issues to confirm the problem, but senders should be aware of the problems this may cause. LashBack recommends all senders and advertisers to use unique unsubscribe mechanisms to match unsubscribe segmentation expectations set forth above. However, it should be reiterated that senders or advertisers that constantly change unsubscribe mechanisms will either have a poor reputation or no reputation and will not qualify for UnsubSafe.sts globally.

© 2003-2007 LashBack, LLC.
555 Washington Ave. Suite 400, St. Louis, MO 63101
(800) 434-1994

https://web.archive.org/web/20090130004220/http://www.lashback.com:80/reputation/

INTERNET ARCHIVE WayBackMachine
http://www.lashback.com/reputation/ Go DEC JAN 30 MAR
46 captures
11 Mar 2007 - 23 Aug 2013
2007 2009 2010
About this capture

reputation
trusted or untrusted?

Reputation
Data Story
Reputation Toolbox
Unsubscribe Best Practices
Blacklist Resources
Feature Comparison Chart

Reputation

Are You Trusted or Untrusted?

A critical part of the LashBack mission is to define the working relationship between compliancy, reputation, deliverability and increased revenue in a best practices context.

The industry standard used to determine deliverability of your email is your reputation. Your reputation takes the form of a score, not unlike a personal credit score and is used by receivers such as Yahoo!, AOL, MSN and all major ISP's to determine whether your email gets delivered. The score is based on a 0 to 100 scale. 100 is perfect and 0 is poor. If you have one case of unsubscribe abuse your score will drop to 0. Even if it not your fault. This shows the need to know who you are working with and who is affecting your business positively with compliance and negatively with non-compliance.

The popular authentication services require a 90 on the 100 point scale. It is difficult, if not impossible to maintain this high reputation score without a comprehensive compliance process.

The unsubscribe portion of your reputation score is determined directly by LashBack and this data, based on the performance of your unsubscribe compliance is shared with receivers through LashBack reputation service partners such as Return Path, Cloudmark, MX Logic, Truste, Unsubscentral and Habeas.

Your reputation has a direct effect on your revenue. If your deliverability suffers from a poor reputation, then your bottom line decreases proportionately. The good news is the reverse is also true. By monitoring your compliancy and adherence to internal standards you can maintain and increase a good reputation. More email delivered means more revenue.

Compliance means reputation and reputation is deliverability.

Unsubscribe, CAN-SPAM and Your Reputation

The CAN-SPAM Act is "opt out" legislation. It does not stop any company from sending anyone compliant commercial email but, if they opt out of your list, it is illegal to send them more email after 10 days. This makes unsubscribe the most critical aspect of CAN-SPAM. The legislative intent of CAN-SPAM as enforced by the FTC is to allow an "email marketplace" where a consumer can opt-out or unsubscribe safely.

LashBack shares this unsubscribe perspective as "The Unsubscribe Authority" of email. As we have grown to become the clear "Email Compliance Authority", LashBack has documented hundreds of unsubscribe mechanisms and processes. UnsubMonitor watches hundreds of thousands of unsub mechanisms and documents their performance through the use of comprehensive, patent-pending technology.

Where to start:

Do you know what your reputation score is?
Do you know the 10 guidelines to check for to ensure CAN-SPAM Compliance?
Do you know what violations you have had?

© 2003-2007 LashBack, LLC.
555 Washington Ave. Suite 400, St. Louis, MO 63101
(800) 434-1994

https://web.archive.org/web/20090213235949/http://www.lashback.com:80/support/ToolbarFAQ.aspx

INTERNET ARCHIVE WayBackMachine
http://www.lashback.com/support/ToolbarFAQ.aspx
Go
DEC FEB MAR
13 captures
17 May 2007 - 24 Feb 2009
2007 2009 2010
About this capture

Feature Comparison Chart

Outlook Toolbar FAQ

Quick Links

About Spam and How to Fight It
How LashBack Works
Setup, Cost and Upgrades
Technical Questions

About Spam and How to Fight It

What is 'Spam' and how does it affect me?

Depending on who you ask, 'Spam' could be defined in dozens of ways. However, the average consumer defines Spam as "unwanted email." This is the definition we've adopted.

Spam is annoying, can lower productivity, threaten the security of your computer and even deliver objectionable or offensive content to your email inbox.

What are my options in dealing with Spam? Which is the best?

While most of the world is simply trying to ignore spam with the "delete" key, others have recognized that spam is becoming unmanageable and grows in quantity the longer you use an email address. Many fight spam with:

- **Blocking** – Your email program or ISP (Internet Service Provider) may allow you to Block specific senders from delivering any email to you. This allows you to block regular senders of spam. Unfortunately, since spammers constantly change the addresses they send from this method only stops the least sophisticated spammers and hardly any spam.
- **Filters** – Your ISP, email provider or private software company can sometimes provide a filtering technology that attempts to identify spam (by keywords) before it arrives in your inbox and diverts that message into a "Junk Mail" or "Spam" folder, where it is later deleted. The problem with this method is that 'good' email is often identified as spam and deleted without your knowledge. These are called "false positives" and are the biggest problem with what would otherwise be the most effective way to combat spam.
- **Opting Out (Unsubscribing)** – Many commercial emails provide a button or link that you can click to Opt-Out of the sender's mailing list so you don't get any messages from that sender in the future. The US government has recently identified this as the best way for its citizens to determine which email they choose to receive and which they choose not to receive. The biggest drawback to this method is that (as of December 2003) some commercial emails either do not have opt-out mechanisms or do not have ones that work. Recent legislation (the CAN-SPAM Act) seeks to ensure that ALL commercial email in the future have working unsubscribe functions. Based on this development, Opting Out will be the best long-term solution to spam.

Which methods does LashBack use to fight against Spam?

LashBack incorporates nearly all of the modern methods in spam-fighting. It gives the option to Block specific senders, allows you to filter future messages based on the information that LashBack members supply, and will allow you to opt-out of spam with the click of a button.

If 'opting-out' of spam is the best long-term solution to spam, why do you also bother to block and filter spam?

We at LashBack know that Opting Out will ultimately result in the least Spam received by its members, but also recognize that Blocking and Filtering can bring a more immediate result to spam-elimination and provide some personal customization that many email users desire. By using all three popular methods (and more), we can ensure that our customers receive the least spam possible.

What about the government and the CAN-SPAM Act? Won't that take care of the Spam problem? Do I really need your software?

The CAN-SPAM Act, which was just recently made law in the United States protects citizens from the tide of spam by requiring commercial emails to have a working "Unsubscribe" button or link in it. Failure to comply would result in criminal charges. A conviction would mean fines and/or prison time. This is a wonderful development in the fight against spam, but it has its problems.

While the details of the enforcement of such a law are being rolled out, you will continue to get spam today. And once the law has truly taken effect, you will still have to MANUALLY opt out – meaning you will have to spend more time clicking on links in spam and filling out forms just to unsubscribe. For most of us, this means that we'll have to spend more time than ever on spam, just to take advantage of the new law.

LashBack AUTOMATICALLY unsubscribes you from messages identified as spam. With one-button, you block the sender, filter future messages from that sender and automatically find their unsubscribe form to fill it out.

Doesn't 'Opting-Out' just let the spammers know my email address is valid

and ultimately get me more Spam?

Some spammers make fake unsubscribe buttons/links and use clicks to determine which email addresses are valid – so they can continue to spam them.

LashBack addresses this possibility with a 'Trust System' which identifies email advertisers as Trusted (someone known to honor Unsubscribe requests), Untrusted (someone who is known to abuse the requests to send more spam), or Unknown.

Advertisers begin as labeled 'Unknown,' during which time your Opt-out requests are recorded by the LashBack servers but are not sent. Instead, LashBack sends out a 'probe' (an opt-out request by the server) to test to determine if the process is safe. If it's safe, the advertiser becomes Trusted and your request is processed. If it is unsafe (they send back spam), they become Untrusted and your request is not sent (you simply block and filter them, while LashBack gathers evidence of their illegal activity). If LashBack cannot determine either way (say, the link is broken), your opt-out request is held while additional testing is conducted.

This Safe Unsubscribe feature is patent-pending.

How LashBack Works

How does LashBack stop spam?

LashBack's approach to fighting spam is a multi-tiered attack. When you highlight a message and click on the red LashBack lightning bolt button on your email client's toolbar, several things all happen at once:

- The message is removed from your inbox and is placed in a custom "Spam" folder.
- The sender of that message is added to your "Blocked" List. Future messages from that sender are prevented from reaching your inbox.
- LashBack gets a copy of the spam message from you, records it, scans it and then (based on the website they're advertising), filters all messages that advertise the same product or service.
- If the advertiser's Unsubscribe mechanism is considered to be Trusted, you are automatically unsubscribed from that spammer's email list.
- The unique and simple combination of these actions eliminates the spam you have today and prevents you from getting spam tomorrow (and on).

If one button does all the work, what do the other three buttons do?

The other three buttons are:

- Add To SafeList - Highlight a message and click this button to add a relative, friend, coworker or other trusted source to a list of addresses which guarantees messages from those email addresses will always be delivered successfully to you.
- Configure - Change your personal options, review your blocked list, initiate a scan for spam in your inbox and much more.
- Reports - Launches our Spam Control program which can let you run reports to determine what messages are being filtered, who's sending them, which ones are honoring your opt-out requests and more.

Does LashBack do anything to discourage spammers from sending me spam in the first place?

In addition to blocking, filtering and unsubscribing you from spam, LashBack also uses the information it gathers on the illegal activities of spammers to work with anti-spam organizations worldwide to find and punish spammers for their criminal activities. In this way, LashBack Community members (subscribers) are all contributing to the end of spam as we know it. This is precisely how you are able to 'lash back' against spam.

Setup, Cost and Upgrades

How hard is it to set LashBack up? How hard is it to use?

LashBack is designed with the Beginner in mind. We figure spam is annoying enough without it becoming MORE complicated to get rid of it. We want using LashBack to be as easy as using the delete key, only a LOT more powerful in the fight against spam.

Installation and setup takes less than five minutes. A quick interactive tutorial is included to go over all of the most commonly-used functions and an online manual covers everything else you might need to know.

How much does LashBack cost to use?

The LashBack toolbar costs absolutely nothing to use. Updates to the service, upgrades to the product and technical support are all included for free.

Does LashBack work with Macintosh or Unix computers?

No, we do not currently support those operating systems. We do, however, have plans to offer these options in the future.

I don't use Outlook or Outlook Express for my email, but I want to try LashBack. Do you have plans to release other editions of the software to cover other email clients like Eudora, MSN, AOL or Hotmail?

Other versions are continually in the works! If you would like to be notified when new versions become available, you can join our mailing list by contacting us at updates@lashback.com with "new versions" as the subject line.

Technical Questions

I have more than one email address. Will LashBack work with all of those email addresses at once, or…?

LashBack protects all email addresses used by one computer (that are configured for use with email clients we support). A separate copy of LashBack is required for additional computers, but not for additional email addresses.

How much of my computer's resources will LashBack be using? Will it affect the other things I already do on my computer?

While the amount of processing power devoted to LashBack is customizable, most users will default to the same amount of processing power and internet bandwidth usage as would be required to view a simple web page. In other words, you are highly unlikely to notice any diminishment in processing power or internet speed.

Will Lashback ever block or filter 'good' email?

While it is unlikely, it is possible that while you begin to use the Filtering offered by LashBack, some email from trusted sources may diverted incorrectly into the Spam folder. If this happens, simply highlight the message in your spam folder and click on "Add to SafeList" so that any message from that email address will always be allowed to go into your inbox.

It is highly recommended that you add all friends, family and other trusted email addresses to your SafeList as soon as you install LashBack to ensure the best possible filtering results.

How can I add someone to my SafeList if I don't already have an email of theirs to highlight?

On the LashBack toolbar, click on "Configure" and choose the "Spam Control" button (alternately, you can double-click the Lashback icon in your system tray). Click on the green SafeList button to access the list of trusted addresses. Then, click on "Add" and type in the email address to be added. Click OK and you're done!

© 2003-2007 LashBack, LLC.
555 Washington Ave. Suite 400, St. Louis, MO 63101
(800) 434-1994

https://web.archive.org/web/20090202193636/http://www.lashback.com:80/ToolbarDownload.aspx

INTERNET ARCHIVE
WayBackMachine
http://www.lashback.com/ToolbarDownload.aspx
Go
JUN FEB AUG
7 captures
6 Sep 2007 - 20 Aug 2010
2007 02 2009 2010
About this capture



© 2003-2009 LashBack, LLC., 555 Washington Ave. Suite 400, St. Louis, MO 63101

https://web.archive.org/web/20110102084641/http://www.constantcontact.com/email-marketing/email-list/index.jsp