

The IMAP SEARCH=ADDRESS Extension
draft-ietf-morg-address-search-00.txt

Status of this Memo

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

Copyright (c) 2009 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.

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>. The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft expires in July 2009.

Abstract

The SEARCH=ADDRESS extension extends IMAP search to operate on exact addresses and domains. Without this extension, one has to search on substrings and/or do the searching clientside.

1. Conventions Used in This Document

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

Formal syntax is defined by [[RFC5234](#)].

Example lines prefaced by "C:" are sent by the client and ones prefaced by "S:" by the server. The five characters [...] means that something has been elided.

2. Overview

This document defines a few extra search keys to search on email addresses qua addresses.

Base IMAP allows only searching on substrings, so searching on addresses involves relying on server-specific behaviour, accepting false positives, accepting false negatives, or all three. This works acceptably when a human wants to find one or a few messages, but less well e.g. when the result set is too large for human inspection.

Substantive comments regarding this draft may be sent to the morg@ietf.org list. Send nits to the author only, please.

3. New Search Keys

This document defines three new search keys that operate on addresses in address fields.

3.1. Address Considerations

In this document, an address is only localpart@domain. Display-name and route information is disregarded, so the address in this example is "nirmala@example.com":

From: Nirmala Devi <@exemplo.com.br:nirmala@example.com>

Some email address parsers accept additional syntax, for example the so-called percent hack to support legacy systems. Such syntax should be treated similarly. Addresses without a domain (usually sent from the local system) are particularly common. These should be provided with an appropriate domain.

Syntactically speaking, any string is valid as argument to these search keys. If the server can determine that the argument cannot match any messages (e.g. ADDRESS "@"), then... open issue. Two alternatives I think: 1. tagged NO with a BADADDRESS response code. 2. Carrying out the search, probably returning an empty search result, and returning a tagged OK with a BADADDRESS response code.

(Open issue: I think the localpart ought to be considered case insensitive. All agree?_)

Open issue: EAI addresses.

3.2. Address Fields

The following fourteen address fields MUST be supported: From, To, Cc, Bcc, Reply-To, Return-Path, Sender, Resent-Bcc, Resent-Cc, Resent-From, Resent-Sender and Resent-To.

The server MAY also support other fields. If it does, then it MUST parse them using the correct syntax for those fields. It is not permissible to parse unknown fields using the syntax for e.g. To, From, or Return-Path.

All three search keys use either a client-specified list of address fields, or all supported fields.

Open issue. What's better?

1. The client MUST NOT specify fields other than those specified above. The only way to test other address fields than the mandatory fourteen is to search all fields.

2. If the client specifies an unsupported field, the server MUST respond with NO.

The server MUST check address fields in the top-level message, and MAY also check header fields in subsidiary message/rfc822 objects. (Open issue: SHOULD also? Or even MUST?)

3.3. The ADDRESS Search Key

The ADDRESS search key matches a messages if one of its address fields refer to the specified address. This command finds all messages to/from/... fred@example.com, without matching messages to/from/... alfred@example.com or fred@example.com.au:

```
C: a UID SEARCH ADDRESS ANY fred@example.com
```

3.4. The DOMAIN Search Key

The DOMAIN search key matches a messages if one of its address fields refer to the specified domain.

This command finds all messages sent by any example.com address, but does not match messages sent by asdf@example.com.au or hostmaster@ns1.example.com.

```
C: a UID SEARCH DOMAIN ("From" "Sender" "Reply-To") example.com
```

3.5. The SUBDOMAIN Search Key

The SUBDOMAIN search key matches a messages if one of its address fields refer to the specified domain, or a subdomain of it the specified domain.

This command finds all messages sent by any example.com or *.example.com address, but not example.com.au:

```
C: a UID SEARCH SUBDOMAIN (From Sender Reply-To) example.com
```

3.5. Client Fallbacks

If an IMAP server does not support ADDRESS, the client may fall back to the FROM, TO, HEADER etc. search keys. The rest of this section uses only FROM, for simplicity.

Matching a localpart can be done either with "FROM <fred@" (which eliminates false positives but may or may not match a field such as "From: fred@example.com") or with "FROM fred@" (which gives false positives for messages "From: alfred@example.com.uy").

Matching a domain (without subdomains) can be done either with "FROM @example.com>" (may miss some results) or "FROM @example.com" (may include false positives such as example.com.uy).

Matching a domain including subdomains has to be done using either "FROM example.com>" (which may miss results and may include false positives such as notexample.com) or "FROM example.com" (which may include false positives such as foo-return-x432y69-example.com@beispiel.de and leon@example.comintern.int).

An IMAP client can achieve exact search results by choosing the overly inclusive search keys, downloading the search results and filtering out false positives. (This will however not find forwarded messages, since FROM and friends only look at the top-level header.)

Note that some widely-used servers process address fields as part of parsing, so that e.g. "From: <fred@example.com>" may be rewritten as "From: fred@example.com" or the other way around. Some of these fallback searches may yield different results on different servers.

4. Formal Syntax

The following syntax specification uses the Augmented Backus-Naur Form (ABNF) notation as specified in [\[RFC5234\]](#). [\[RFC3501\]](#) defines the non-terminals "capability", "search-key", and "astring". [\[RFC5322\]](#) defines "field-name".

Except as noted otherwise, all alphabetic characters are case-insensitive. The use of upper or lower case characters to define token strings is for editorial clarity only. Implementations MUST accept these strings in a case-insensitive fashion.

```
capability    =/ "SEARCH=ADDRESS"

search-key    =/ ("ADDRESS" / "DOMAIN" / "SUBDOMAIN" )
               [ SP "(" address-field *(SP address-field) ")" ]
               SP astring

address-field = "From" / "To" / "Cc" / "Bcc" / "Reply-To" /
               "Return-Path" / "Sender" / "Resent-Bcc" /
               "Resent-Cc" / "Resent-From" / "Resent-Sender" /
               "Resent-To" / optional-address-field

optional-address-field = field-name
```

Open issue: field-name includes ' ')'. It is rather too inclusive. My preferred solution is to do away with optional-address-field. I can't think of any use cases that need it anyway.

5. Security Considerations

This document is believed not to have any security implications.

6. IANA Considerations

The IANA is requested to add SEARCH=ADDRESS to the list of IMAP extensions, <http://www.iana.org/assignments/imap4-capabilities>.

7. Acknowledgements

Go ahead, make me add you.

8. Normative References

- [RFC2119] Bradner, "Key words for use in RFCs to Indicate Requirement Levels", [RFC 2119](#), Harvard University, March 1997.
- [RFC3501] Crispin, "Internet Message Access Protocol - Version 4rev1", [RFC 3501](#), University of Washington, June 2003.
- [RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", [RFC 5234](#), January 2008.
- [RFC5322] Resnick, "Internet Message Format", [RFC 5322](#), Qualcomm, October 2008.

9. Author's Address

Arnt Gulbrandsen
Oryx Mail Systems GmbH
Schweppermannstr. 8
D-81671 Muenchen
Germany

Fax: +49 89 4502 9758

Email: arnt@oryx.com

(RFC Editor: Please delete everything after this point)

Changes since -00

- ...