

Message Organization Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: May 20, 2011

T. Sirainen  
November 16, 2010

IMAP4 Extension for Fuzzy Search  
draft-ietf-morg-fuzzy-search-03

Abstract

This document describes an IMAP protocol extension enabling server to perform searches with inexact matching and assigning relevancy scores for matched messages.

Note

A revised version of this draft document will be submitted to the RFC editor as a Proposed Standard for the Internet Community. Discussion and suggestions for improvement are requested, and should be sent to [morg@ietf.org](mailto:morg@ietf.org).

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on May 20, 2011.

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.

## 1. Conventions used in this document

In examples, "C:" indicates lines sent by a client that is connected to a server. "S:" indicates lines sent by the server to the client.

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 [Kwds].

## 2. Introduction

When humans perform searches in IMAP clients, they typically want to see the most relevant search results first. IMAP servers are able to do this in the most efficient way when they're free to internally decide how searches should match messages. This document describes a new SEARCH=FUZZY extension that provides such functionality.

## 3. The FUZZY Search Key

FUZZY search key takes another search key as its argument. Server is allowed to perform all matching in an implementation-defined manner for this search key, including ignoring the active comparator as defined by [RFC5255]. Typically this would be used to search for strings, for example:

```
C: A1 SEARCH FUZZY (SUBJECT "IMAP break")
S: * SEARCH 1 5 10
S: A1 OK Search completed.
```

Besides matching messages with subject "IMAP break", the above search may also match messages with subjects "broken IMAP", "IMAP is broken", or anything else the server decides that might be a good match.

This example does a fuzzy SUBJECT search, but a non-fuzzy FROM search:

```
C: A2 SEARCH FUZZY SUBJECT work FROM user@example.com
S: * SEARCH 1 4
```

S: A2 OK Search completed.

It is implementation-defined how server handles multiple separate FUZZY search keys.

#### 4. Relevancy Scores for Search Results

Servers SHOULD assign a search relevancy score for each matched message when the FUZZY search key is given. Relevancy scores are given in range 1-100, where 100 is the highest relevancy. The relevancy scores SHOULD use the full 1-100 range, so that clients can show them to users in a meaningful way, such as a percentage value.

As the name already tells, relevancy scores specify how relevant to the search the matched message is. It's not necessarily the same as how precisely the message matched. For example a message whose subject matches fuzzily the search string might get a higher relevancy score than a message whose body had the exact string in the middle of a sentence. When multiple search keys are matched fuzzily, it's server-dependent on how the relevancy score is calculated.

If server also advertises the ESEARCH capability as defined by [ESEARCH], the relevancy scores can be retrieved using the new RELEVANCY return option for SEARCH:

```
C: B1 SEARCH RETURN (RELEVANCY ALL) FUZZY TEXT "Helo"
S: * ESEARCH (TAG "B2") ALL 1,5,10 RELEVANCY (4 99 42)
S: B1 OK Search completed.
```

The RELEVANCY return option MUST NOT be used unless FUZZY search key is also given. Note that SEARCH results aren't sorted by relevancy, SORT is needed for that.

#### 5. Fuzzy matching with non-string search keys

Fuzzy matching is not limited to just string matching. All search keys SHOULD be matched fuzzily, although what exactly that means for different search keys is left up to server implementations to decide -- including deciding that fuzzy matching is meaningless for a particular key, and falling back to exact matching. Some suggestions are given below.

Dates: A typical example could be when a user wants to find a message "from Dave about a week ago". A client could perform this search using SEARCH FUZZY (FROM "Dave" SINCE 21-Jan-2009 BEFORE 24-Jan-2009). Server could return messages outside the specified date

range, but the further away the message is, the lower the relevancy score.

Sizes: These should be handled similar to dates. If a user wants to search for "about 1 MB attachments", the client could do this by sending SEARCH FUZZY (LARGER 900000 SMALLER 1100000). Again the further away the message size is from the specified range, the lower the relevancy score.

Flags: Server could return messages that don't have the specified flags, but with a lower relevancy score.

UIDs, sequences, modification sequences: These are examples of keys for which exact matching is probably what makes sense. Alternatively, a server might choose, for instance, to expand a UID range by 5% on each side.

## 6. Extensions to SORT

If server also advertises the SORT capability as defined by [SORT], the results can be sorted by the new RELEVANCY sort criteria:

```
C: C1 SORT (RELEVANCY) UTF-8 FUZZY SUBJECT "Helo"
S: * SORT 5 10 1
S: C1 OK Sort completed.
```

The message with the highest score is returned first. As with RELEVANCY return option, RELEVANCY sort criteria MUST NOT be used unless FUZZY search key is also given.

If server also advertises the ESORT capability as defined by [CONTEXT], the relevancy scores can be retrieved using the new RELEVANCY return option for SORT:

```
C: C2 SORT RETURN (RELEVANCY ALL) (RELEVANCY) UTF-8 FUZZY TEXT "Helo"
S: * ESEARCH (TAG "C2") ALL 5,10,1 RELEVANCY (99 42 4)
S: C2 OK Sort completed.
```

To limit the number of returned messages, use the PARTIAL return option. For example this returns the 10 most relevant messages:

```
C: C3 SORT RETURN (PARTIAL 1:10) (RELEVANCY) UTF-8 FUZZY TEXT "World"
S: * ESEARCH (TAG "C3") PARTIAL (1:10 42,9,34,13,15,4,2,7,23,82)
S: C3 OK Sort completed.
```

## 7. Formal Syntax

The following syntax specification uses the augmented Backus-Naur Form (BNF) as described in [ABNF]. It includes definitions from [RFC3501], [IMAP-ABNF] and [SORT].

```
capability          =/ "SEARCH=FUZZY"

score               = 1*3DIGIT
                    ;; (1 <= n <= 100)

score-list          = "(" [score *(SP score)] ")"

search-key          =/ "FUZZY" SP search-key

search-return-data  =/ "RELEVANCY" SP score-list
                    ;; Conforms to <search-return-data>, from [IMAP-ABNF]

search-return-opt   =/ "RELEVANCY"
                    ;; Conforms to <search-return-opt>, from [IMAP-ABNF]

sort-key            =/ "RELEVANCY"
```

## 8. Security Considerations

Implementation of this extension might enable a denial-of-service attack if the implementation isn't careful to prevent them. Fuzzy search engines are often complex with non-obvious disk space, memory and/or CPU usage patterns. Implementors should test at least the behavior of large messages that contain very long words and/or unique random strings. Also very long search keys might cause excessive memory or CPU usage.

Invalid input may also be problematic. For example if the search engine takes UTF-8 stream as input, it might fail more or less badly when illegal UTF-8 sequences are fed to it from a message whose character set was claimed to be UTF-8. This could be avoided by validating all the input and, for example, replacing illegal UTF-8 sequences with the Unicode replacement character (U+FFFD).

Search relevancy rankings might be susceptible to "poisoning" by smart attackers using certain keywords or hidden markup (e.g. HTML) in their messages to boost the rankings. This can't be fully prevented by servers, so clients should prepare for it by at least allowing user to see all the search results, rather than hide results below a certain score.

## 9. IANA Considerations

IMAP4 capabilities are registered by publishing a standards track or IESG approved experimental RFC. The registry is currently located at:

<http://www.iana.org/assignments/imap4-capabilities>

This document defines the X-DRAFT-I03-SEARCH=FUZZY [[anchor7: Note to RFC Editor: fix before publication]] IMAP capability. IANA is requested to add it to the registry.

## 10. Acknowledgements

Alexey Melnikov, Zoltan Ordogh, Barry Leiba, Cyrus Daboo and Dave Cridland have helped with this document.

## 11. Normative References

- [ABNF] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 5234, January 2008.
- [CONTEXT] Cridland, D. and C. King, "Contexts for IMAP4", RFC 5267, July 2008.
- [ESEARCH] Melnikov, A. and D. Cridland, "IMAP4 Extension to SEARCH Command for Controlling What Kind of Information Is Returned", RFC 4731, November 2006.
- [IMAP-ABNF] Melnikov, A. and C. Daboo, "Collected Extensions to IMAP4 ABNF", RFC 4466, April 2006.
- [Kwds] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.
- [RFC3501] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", RFC 3501, March 2003.
- [RFC5255] Newman, C., Gulbrandsen, A., and A. Melnikov, "Internet Message Access Protocol Internationalization", RFC 5255, June 2008.
- [SORT] Crispin, M. and K. Murchison, "Internet Message Access

Protocol - SORT and THREAD Extensions", RFC 5256,  
June 2008.

Author's Address

Timo Sirainen

Email: [tss@iki.fi](mailto:tss@iki.fi)





Message ORGanization Working Group  
Internet-Draft  
Updates: 4466 (if approved)  
Intended status: Experimental  
Expires: September 6, 2011

B. Leiba  
Huawei Technologies  
A. Melnikov  
Isode Limited  
March 5, 2011

IMAP4 Multimailbox SEARCH Extension  
draft-ietf-morg-multimailbox-search-07

Abstract

The IMAP4 specification allows the searching only of the selected mailbox. A user often wants to search multiple mailboxes, and a client that wishes to support this must issue a series of SELECT and SEARCH commands, waiting for each to complete before moving on to the next. This extension allows a client to search multiple mailboxes with one command, limiting the round-trips and waiting for various searches to complete, and not requiring disruption of the currently selected mailbox. This also uses MAILBOX and TAG fields in ESEARCH responses, allowing a client to pipeline the searches if it chooses.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on September 6, 2011.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1.	Introduction . . . . .	3
1.1.	Conventions used in this document . . . . .	3
2.	New ESEARCH command . . . . .	3
2.1.	The ESEARCH response . . . . .	4
2.2.	Source options: specifying mailboxes to search . . . . .	5
3.	Examples . . . . .	7
4.	Formal Syntax . . . . .	7
5.	Security Considerations . . . . .	8
6.	IANA Considerations . . . . .	9
7.	Acknowledgements . . . . .	9
8.	Normative References . . . . .	9
	Authors' Addresses . . . . .	10

## 1. Introduction

The IMAP4 specification allows the searching only of the selected mailbox. A user often wants to search multiple mailboxes, and a client that wishes to support this must issue a series of SELECT and SEARCH commands, waiting for each to complete before moving on to the next. The commands can't be pipelined, because the server might run them in parallel, and the untagged SEARCH responses could not then be distinguished from each other.

This extension allows a client to search multiple mailboxes with one command, and includes MAILBOX and TAG fields in the ESEARCH response, yielding the following advantages:

- o A single command limits the number of round-trips needed to search a set of mailboxes.
- o A single command eliminates the need to wait for one search to complete before starting the next.
- o A single command allows the server to optimize the search, if it can.
- o A command that is not dependent upon the selected mailbox eliminates the need to disrupt the selection state, or to open another IMAP connection.
- o The MAILBOX and TAG fields in the responses allow a client to distinguish which responses go with which search (and which mailbox). A client can safely pipeline these search commands without danger of confusion.

### 1.1. Conventions used in this document

In examples, "C:" indicates lines sent by a client that is connected to a server. "S:" indicates lines sent by the server to the client.

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 [RFC2119].

## 2. New ESEARCH command

Arguments: OPTIONAL source options  
 OPTIONAL result options  
 OPTIONAL charset specification (see [RFC2978])  
 searching criteria (one or more)

Responses: REQUIRED untagged response: ESEARCH

Result: OK - search completed  
 NO - error: cannot search that charset or criteria  
 BAD - command unknown or arguments invalid

This section defines a new ESEARCH command, which works similarly to the UID SEARCH command described in section 2.6.1 of [RFC4466] (initially described in section 6.4.4 of [RFC3501] and extended by [RFC4731]).

The ESEARCH command further extends searching by allowing for optional source and result options. This document does not define any new result options (see Section 3.1 of [RFC4731]). A server that supports this extension includes "MULTISEARCH" in its IMAP capability string.

Because there has been confusion about this, it is worth pointing out that with ESEARCH, as with \*any\* SEARCH or UID SEARCH command, it MUST NOT be considered an error if the search terms include a range of message numbers that extends (or, in fact, starts) beyond the end of the mailbox. For example, a client might want to establish a rolling window through the search results this way:

```
C: tag1 UID ESEARCH FROM "frobozz" 1:100
```

...followed later by this:

```
C: tag1 UID ESEARCH FROM "frobozz" 101:200
```

...and so on. This tells the server to match only the first hundred messages in the mailbox the first time, the second hundred the second time, etc. In fact, it might likely allow the server to optimize the search significantly. In the above example, whether the mailbox contains 50 or 150 or 250 messages, neither of the search commands shown will result in an error. It is up to the client to know when to stop moving its search window.

## 2.1. The ESEARCH response

In response to an ESEARCH command, the server MUST return ESEARCH responses [RFC4731] (that is, not SEARCH responses). Because message numbers are not useful for mailboxes that are not selected, the

responses MUST contain information about UIDs, not message numbers. This is true even if the source options specify that only the selected mailbox be searched.

Presence of a source option in absence of a result option implies the "ALL" result option (see Section 3.1 of [RFC4731]). Note that this is not the same as the result from the SEARCH command described in the IMAP base protocol.

Source options describe which mailboxes must be searched for messages. An ESEARCH command with source options does not affect which mailbox, if any, is currently selected, regardless of which mailboxes are searched.

For each mailbox satisfying the source options, a single ESEARCH response MUST be returned if any messages in that mailbox match the search criteria. An ESEARCH response MUST NOT be returned for mailboxes that contain no matching messages. This is true even when result options such as MIN, MAX, and COUNT are specified (see section 3.1 of [RFC4731]), and the values returned (lowest UID matched, highest UID matched, and number of messages matched, respectively) apply to the mailbox reported in that ESEARCH response.

Note that it is possible for an ESEARCH command to return \*no\* untagged responses (no ESEARCH responses at all), in the case that there are no matches to the search in any of the mailboxes that satisfy the source options. Clients can detect this situation by finding the tagged OK response without having received any matching untagged ESEARCH responses.

Each ESEARCH response MUST contain the MAILBOX, TAG, and UIDVALIDITY correlators. Correlators allow clients to issue several ESEARCH commands at once (pipelined). If the SEARCHRES [RFC5182] extension is used in an ESEARCH command, that ESEARCH command MUST be executed by the server after all previous SEARCH/ESEARCH commands have completed, and before any subsequent SEARCH/ESEARCH commands are executed. The server MAY perform consecutive ESEARCH commands in parallel as long as none of them use the SEARCHRES extension.

## 2.2. Source options: specifying mailboxes to search

The source options, if present, MUST contain a mailbox specifier as defined in the IMAP NOTIFY extension [RFC5465], section 6 (using the "filter-mailboxes" ABNF item), with the following differences:

1. The "selected-delayed" specifier is not valid here.

2. A "subtree-one" specifier is added. The "subtree" specifier results in a search of the specified mailbox and all selectable mailboxes that are subordinate to it, through an indefinitely deep hierarchy. The "subtree-one" specifier results in a search of the specified mailbox and all selectable child mailboxes, one hierarchy level down.

If "subtree" is specified, the server MUST defend against loops in the hierarchy (for example, those caused by recursive file-system links within the message store). The server SHOULD do this by keeping track of the mailboxes that have been searched, and terminating the hierarchy traversal when a repeat is found. If it cannot do that, it MAY do it by limiting the hierarchy depth.

If the source options are not present, the value "selected" is assumed -- that is, only the currently selected mailbox is searched.

The "personal" source option is a particularly convenient way to search all of the current user's mailboxes. Note that there is no way to use wildcard characters to search all mailboxes; the "mailboxes" source option does not do wildcard expansion.

If the source options include (or default to) "selected", the IMAP session MUST be in "selected" state. If the source options specify other mailboxes and NOT "selected", then the IMAP session MUST be in either "selected" or "authenticated" state. If the session is not in a correct state, the ESEARCH command MUST return a "BAD" result.

If the server supports the SEARCHRES [RFC5182] extension, then the "SAVE" result option is valid *only* if "selected" is specified or defaulted as the sole mailbox to be searched. If any source option other than "selected" is specified, the ESEARCH command MUST return a "BAD" result.

If the server supports the CONTEXT=SEARCH and/or CONTEXT=SORT extension [RFC5267], then the following additional rules apply:

- o The CONTEXT return option (Section 4.2 of [RFC5267]) can be used with an ESEARCH command.
- o If the UPDATE return option is used (Section 4.3 of [RFC5267]), it MUST apply *ONLY* to the the currently selected mailbox. If UPDATE is used and there is no mailbox currently selected, the ESEARCH command MUST return a "BAD" result.
- o The PARTIAL search return option (Section 4.4 of [RFC5267]) can be used and applies to each mailbox searched by the ESEARCH command.

If the server supports the ACL [RFC4314] extension, then the logged in user is required to have the 'r' right for each mailbox she wants to search. In addition, any mailboxes that are not explicitly named (accessed through "personal" or "subtree", for example) are required to have the "l" right. Mailboxes matching the source options for which the logged in user lacks sufficient rights MUST be ignored by the ESEARCH command processing. In particular, ESEARCH responses MUST NOT be returned for those mailboxes.

### 3. Examples

In the following example, note that two ESEARCH commands are pipelined, and that the server is running them in parallel, interleaving a response to the second search amid the responses to the first (watch the tags).

```
C: tag1 ESEARCH IN (mailboxes "folder1" subtree "folder2") unseen
C: tag2 ESEARCH IN (mailboxes "folder1" subtree-one "folder2")
subject "chad"
S: * ESEARCH (TAG "tag1" MAILBOX "folder1" UIDVALIDITY 1) UID ALL
4001,4003,4005,4007,4009
S: * ESEARCH (TAG "tag2" MAILBOX "folder1" UIDVALIDITY 1) UID ALL
3001:3004,3788
S: * ESEARCH (TAG "tag1" MAILBOX "folder2/banana" UIDVALIDITY 503)
UID ALL 3002,4004
S: * ESEARCH (TAG "tag1" MAILBOX "folder2/peach" UIDVALIDITY 3) UID
ALL 921691
S: tag1 OK done
S: * ESEARCH (TAG "tag2" MAILBOX "folder2/salmon" UIDVALIDITY
1111111) UID ALL 50003,50006,50009,50012
S: tag2 OK done
```

### 4. Formal Syntax

The following syntax specification uses the Augmented Backus-Naur Form (ABNF) as described in [RFC5234]. Terms not defined here are taken from [RFC3501], [RFC5465], or [RFC4466].

```
command-auth =/  esearch
                ; Update definition from IMAP base [RFC3501]
                ; Add new "esearch" command.
```

```
command-select =/ esearch
                ; Update definition from IMAP base [RFC3501]
                ; Add new "esearch" command.
```

```
filter-mailboxes-other =/ ("subtree-one" SP one-or-more-mailbox)
    ; Update definition from IMAP Notify [RFC5465]
    ; Add new "subtree-one" selector.

filter-mailboxes-selected = "selected"
    ; Update definition from IMAP Notify [RFC5465]
    ; We forbid the use of "selected-delayed".

one-correlator = ("TAG" SP tag-string) / ("MAILBOX" SP astring) /
    ("UIDVALIDITY" SP nz-number)
    ; Each correlator MUST appear exactly once

scope-option = scope-option-name [SP scope-option-value]
    ; No options defined here. Syntax for future extensions.

scope-option-name = tagged-ext-label
    ; No options defined here. Syntax for future extensions.

scope-option-value = tagged-ext-val
    ; No options defined here. Syntax for future extensions.

scope-options = scope-option *(SP scope-option)
    ; A given option may only appear once
    ; No options defined here. Syntax for future extensions.

esearch = "ESEARCH" [SP esearch-source-opts]
    [SP search-return-opts] SP search-program

search-correlator = SP "(" one-correlator *(SP one-correlator) ")"
    ; Updates definition in IMAP4 ABNF [RFC4466]

esearch-source-opts = "IN" SP "(" source-mbox [SP
    "(" scope-options ")" ] ")"

source-mbox = filter-mailboxes *(SP filter-mailboxes)
    ; filter-mailboxes is defined in IMAP Notify [RFC5465]
    ; See updated definition of filter-mailboxes-other, above.
    ; See updated definition of filter-mailboxes-selected, above.
```

## 5. Security Considerations

This new IMAP ESEARCH command allows a single command to search many mailboxes at once. On the one hand, a client could do that by sending many IMAP SEARCH commands. On the other hand, this makes it easier for a client to overwork a server, by sending a single command that results in an expensive search of tens of thousands of mailboxes. Server implementations need to be aware of that, and



provide mechanisms that prevent a client from adversely affecting other users. Limitations on the number of mailboxes that may be searched in one command, and/or on the server resources that will be devoted to responding to a single client, are reasonable limitations for an implementation to impose.

Implementations MUST, of course, apply access controls appropriately, limiting a user's access to ESEARCH in the same way its access is limited for any other IMAP commands. This extension has no data-access risks beyond what may be there in the unextended IMAP implementation.

Mailboxes matching the source options for which the logged in user lacks sufficient rights MUST be ignored by the ESEARCH command processing (see the paragraph about this in Section 2.2). In particular, any attempt to distinguish insufficient access from non-existent mailboxes may expose information about the mailbox hierarchy that isn't otherwise available to the client.

If "subtree" is specified, the server MUST defend against loops in the hierarchy (see the paragraph about this in Section 2.2).

## 6. IANA Considerations

IMAP4 capabilities are registered by publishing a standards track or IESG approved experimental RFC. The registry is currently located here:

<http://www.iana.org/assignments/imap4-capabilities>

This document defines the IMAP capability "MULTISEARCH" and IANA is asked to add it to the registry.

## 7. Acknowledgements

The authors gratefully acknowledge feedback provided by Timo Sirainen, Peter Coates and Arnt Gulbrandsen.

## 8. Normative References

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

- [RFC2978] Freed, N. and J. Postel, "IANA Charset Registration Procedures", BCP 19, RFC 2978, October 2000.
- [RFC3501] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", RFC 3501, March 2003.
- [RFC4314] Melnikov, A., "IMAP4 Access Control List (ACL) Extension", RFC 4314, December 2005.
- [RFC4466] Melnikov, A. and C. Daboo, "Collected Extensions to IMAP4 ABNF", RFC 4466, April 2006.
- [RFC4731] Melnikov, A. and D. Cridland, "IMAP4 Extension to SEARCH Command for Controlling What Kind of Information Is Returned", RFC 4731, November 2006.
- [RFC5182] Melnikov, A., "IMAP Extension for Referencing the Last SEARCH Result", RFC 5182, March 2008.
- [RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.
- [RFC5267] Cridland, D. and C. King, "Contexts for IMAP4", RFC 5267, July 2008.
- [RFC5465] Gulbrandsen, A., King, C., and A. Melnikov, "The IMAP NOTIFY Extension", RFC 5465, February 2009.

#### Authors' Addresses

Barry Leiba  
Huawei Technologies

Phone: +1 646 827 0648  
Email: [barryleiba@computer.org](mailto:barryleiba@computer.org)  
URI: <http://internetmessagingtechnology.org/>

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

Email: [Alexey.Melnikov@isode.com](mailto:Alexey.Melnikov@isode.com)  
URI: <http://www.melnikov.ca/>

