

Internet Draft: Reporting flag state in IMAP
Document: [draft-melnikov-imap-createparams-00.txt](#)
Expires: January 2006
Intended category: Standards Track

A. Melnikov
Isode Ltd.
July 2005

IMAP CREATE parameters

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

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>.

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 the IMAPEXT Mailing list <ietf-imapext@imc.org>. Distribution of this draft is unlimited.

Abstract

When creating (or renaming) a mailbox in [\[IMAP4\]](#) it is desirable to be able to specify additional creation time parameters that can't be changed after the mailbox is created. Some examples of the creation time parameters are: mailbox type, mailbox location on a server or a cluster of servers, mailbox flag state. This document extends IMAP CREATE and RENAME commands to allow for such parameters.

A server which supports this extension indicates this with a capability name of "X-DRAFT-I00-CREATEPARAM".

Table of Contents

0.	To do.....	1
1.	Conventions Used in this Document.....	2
2.	Extended CREATE command.....	X
2.1.	Partition identifier parameter to CREATE command.....	X
2.2.	TYPE parameter to CREATE command.....	X
2.3.	SHAREDFLAGS parameter to CREATE command.....	X
3.	Extended RENAME command.....	X
4.	Security Considerations.....	X
5.	Formal Syntax.....	X
6.	IANA considerations.....	X
7.	Acknowledgments.....	X
8.	Normative References.....	X
9.	Author's Addresses.....	X
10.	Intellectual Property.....	X
11.	Full Copyright Statement.....	X

[1.](#) Conventions Used in this Document

"C:" and "S:" in examples show lines sent by the client and server respectively.

The keywords "MUST", "MUST NOT", "SHOULD", "SHOULD NOT", and "MAY" in this document when typed in uppercase are to be interpreted as defined in "Key words for use in RFCs to Indicate Requirement Levels" [[KEYWORDS](#)].

2. Extended CREATE command

Arguments: mailbox name
 OPTIONAL partition identifier
 OPTIONAL list of CREATE parameters

Responses: no specific responses for this command

Result: OK - create completed
 NO - create failure: can't create mailbox with that name
 and parameters
 BAD - argument(s) invalid

See section 6.3.3 of [[IMAP4](#)] for the description of the basic CREATE command. The text in this section only describes how this behavior is modified by additional parameter.

This extension adds the ability to include one or more parameters with the IMAP CREATE command, to turn on or off certain standard behaviour, or to add new optional behaviours required for a particular extension. Optional parameters to the CREATE command are added as a parenthesised list of attribute/value pairs after the mailbox name. Each value can be either an atom, a string or a list. The order of individual parameters is arbitrary. Individual parameters may consist of one or more atoms or strings in a specific order. If a parameter consists of more than one atom or string, it **MUST** appear in its own parenthesised list. Any parameter not defined by extensions that the server supports **MUST** be rejected with a BAD response.

The mailbox name may be followed by an optional partition identifier parameter, which is described in more details in the following section. Servers that don't support the partition concept **SHOULD** ignore the partition parameter.

<<The reason for having a special syntax for "partition" is to have a backward compatibility with the currently deployed Cyrus IMAP server syntax.>>

(Note, If the partition identifier parameter is not specified, the server supporting multiple partitions uses internal policy to assign the new mailbox to one of the existing partitions.)

If one or more of the superior hierarchical mailboxes doesn't exist the server **SHOULD** create such superior mailboxes, as described in section 6.3.3 of [[IMAP4](#)]. Such superior mailboxes **SHOULD** be assigned to the same partition as the mailbox itself. In other words, an attempt to create "foo/bar/zap" on a server in which "/" is the

hierarchy separator character SHOULD create foo/ and foo/bar/ if they do not already exist. If the mailbox "foo" already exists and is assigned to "partition1" and the client requests to create "foo/bar/zap" on "partition2", then the server SHOULD create both "foo/bar" and "foo/bar/zap" on "partition2".

Extended CREATE can't be used to modify parameters of an already existing mailbox. The server MUST return NO to any such request.

Example: C: a CREATE Forms Server1:partition5 (TYPE CALENDAR)
S: a OK CREATE complete

In the above example, the mailbox "Forms" is create on partition names "Server1:partition5" and has a type CALENDAR. See also [section 2.2](#).

[2.1.](#) Partition identifier parameter to CREATE and RENAME commands

Several existing IMAP servers support a concept of "partition". A partition describes a collection of related mailboxes in a mailstore. Each partition is identified by a unique partition identifier, which may contains a globally unique prefix (e.g. host name or domain name), followed by a local partition identifier. For example, a server implementation that stores mailboxes in a filesystem may choose to use the root directory for a partition as the local partition identifier. (See also the Security Considerations section for discussions about partition naming.)

[2.2.](#) TYPE parameter to CREATE command

Many existing IMAP servers provide access to specialized mailstores, for example mailstores that can store voice messages as described in VPIM <<add informative reference>>.

The TYPE parameter to CREATE command allows the client to give a hint about intended usage of the mailbox to be created. Such hint can be used by the server to choose storage format. For example, some storage formats can only store or be optimized for certain types of MIME messages.

This document defines the following initial set of mailbox types:

- "CONTACT" - can contain MIME messages containing text/plain (?) and vCARD
- "CALENDAR" - can contain vCALENDAR objects as described in <<>>
- "VOICE"
- "IMAGE"

- "VIDEO"

<<TODO: review the list and potentially add more types, like task lists>>

2.3. SHAREDFLAGS parameter to CREATE command

Let's call a flag shared for a mailbox if the mailbox may be set up so that any changes to this flag by a user A are persistent and visible to a different user B. Note, that different mailboxes may have different flags as shared.

SHAREDFLAGS parameter allows to specify which system flags and user defined keywords should be shared for the mailbox. It also allows to "precreate" some user defined keyword.

The server is not required to be able to store any particular system flag or user defined keyword as shared. If the server is unable to persistently store certain flags from SHAREDFLAGS list or store certain flags as shared (or unable to store any user defined flag, when a user defined flag is specified), it MUST return NO to the CREATE command with the SHAREDFLAGS parameter and MUST NOT create the mailbox.

If multiple flags are specified in the SHAREDFLAGS parameter the server MUST either be able to store all requested flags as shared or fail the command with the tagged NO response.

The server MAY restrict which users can create a mailbox with SHAREDFLAGS parameter and which flags may be stored as shared.

When a child submailbox is created and no SHAREDFLAGS parameter is specified, the parent SHAREDFLAGS settings SHOULD be used.

If a mailbox created with SHAREDFLAGS parameter is subsequently renamed, the SHAREDFLAGS settings SHOULD be preserved. <<SHOULD because it can be moved to a different namespace that might have different restrictions>>

A server which supports the SHAREDFLAGS parameter to the CREATE command indicates this with a capability name of "X-DRAFT-I00-CREATEFLAGS". This is in addition to the "X-DRAFT-I00-CREATEPARAM" capability.

Example: C: a CREATE Forms (SHAREDFLAGS (\Seen \$MDNSent))
 S: a OK CREATE completed. Requested flags are shared.
 ...


```
C: b SELECT Forms
S: * 172 EXISTS
S: * 1 RECENT
S: * OK [UNSEEN 12] Message 12 is first unseen
S: * OK [UIDVALIDITY 3857529045] UIDs valid
S: * OK [UIDNEXT 4392] Predicted next UID
S: * FLAGS (\Answered \Flagged \Deleted \Seen \Draft
$MDNSent)
S: * OK [PERMANENTFLAGS (\Answered \Flagged \Deleted
\Seen $MDNSent)] Draft flag is not permanent
S: * OK [SHAREDFLAGS (\Seen $MDNSent)] Limited
S: b OK [READ-WRITE] SELECT completed
```

3. Extended RENAME command

Arguments: existing mailbox name
new mailbox name
OPTIONAL new partition identifier
OPTIONAL list of RENAME parameters

Responses: no specific responses for this command

Result: OK - rename completed
NO - rename failure: can't rename mailbox with that
name, can't rename to mailbox with that name,
can't move the mailbox to the specified partition,
etc.
BAD - argument(s) invalid

See section 6.3.5 of [\[IMAP4\]](#) for the description of the basic RENAME command. The text in this section only describes how this behavior is modified by additional parameter.

This extension adds the ability to include one or more parameters with the IMAP RENAME command, to turn on or off certain standard behaviour, or to add new optional behaviours required for a particular extension. Optional parameters to the RENAME command are added as a parenthesised list of attribute/value pairs after the mailbox name. Each value can be either an atom, a string or a list. The order of individual parameters is arbitrary. Individual parameters may consist of one or more atoms or strings in a specific order. If a parameter consists of more than one atom or string, it MUST appear in its own parenthesised list. Any parameter not defined by extensions that the server supports MUST be rejected with a BAD response.

Note that not all CREATE parameters are allowed as RENAME parameters and vice versa.

The RENAME command changes the name of a mailbox from "existing mailbox name" to "new mailbox name".

The new mailbox name may be followed by an optional new partition identifier parameter, which is described in more details in [section 2.1](#). Servers that don't support the partition concept SHOULD ignore the partition parameter. When the new partition identifier parameter is specified the server is requested to "move" the mailbox to a different partition. Thus in order to move a mailbox between two partitions the client can issue a RENAME command with the new mailbox name being the same as the existing mailbox name, and the partition identifier parameter specifying the new partition.

<<The reason for having a special syntax for "partition" is to have a backward compatibility with the currently deployed Cyrus IMAP server syntax.>>

If one or more of the superior hierarchical mailboxes for the new mailbox name doesn't exist the server SHOULD create such superior mailboxes, as described in section 6.3.5 of [\[IMAP4\]](#). Such superior mailboxes SHOULD be assigned to the same partition as the new mailbox name itself. In other words, an attempt to rename "foo/bar/zap" to "baz/rag/zowie" on a server in which "/" is the hierarchy separator character SHOULD create baz/ and baz/rag/ if they do not already exist. If the mailbox "baz" already exists and is assigned to "partition1" and the client requests to move "foo/bar/zap" to "partition2", then the server SHOULD create both "baz/rag" and "baz/rag/zowie" on "partition2".

If the existing mailbox name has inferior hierarchical mailboxes, then the inferior hierarchical mailboxes MUST also be renamed. For example, a rename of "foo" to "zap" will rename "foo/bar" (assuming "/" is the hierarchy delimiter character) to "zap/bar". If the new partition identifier parameter is specified, then all inferior mailboxes SHOULD be moved to the specified partition.

[4.](#) Security Considerations

<<Certain CREATE/RENAME parameters shouldn't be allowed for all users. For example, a server implementation may restrict usage of the partition identifier parameter in CREATE/RENAME to users with administrative privileges.>>

A partition name may disclose too much information about particular implementation. For example, if different partitions are implemented as different directories in a file system, and a

partition name is the file system path, partition name may disclose the file system layout.

5. Formal Syntax

Formal syntax is defined using ABNF [ABNF] as modified by [IMAP4]. Non-terminals referenced but not defined below are as defined by [ABNF], [IMAP4] or [IMAPABNF].

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      =/ "X-DRAFT-IMAP-CREATEPARAM" /
                  "X-DRAFT-IMAP-CREATEFLAGS"
                  ;;capability is defined in [IMAP4]

create          = "CREATE" SP mailbox [SP partition]
                  [SP create_params]
                  ; Use of INBOX gives a NO error

create_params   = "(" create_param *( SP create_param ) ")"

create_param_name = atom
                  ;; IMAPABNF tagged-ext-label?

create_param     = create_param_name SP create_param_value

create_param_value = astring / "(" astring *(SP astring) ")"
                  ;; As SELECT parameters in ANNOTATE.
                  ;; <<This should be updated to match
                  ;; IMAPABNF>>

partition       = [partition_server ":" ] partition_local
                  ;; use astring instead?

partition_server = atom
                  ;; No ":" allowed, unless IPv6 address?

partition_local  = atom
                  ;; No ":" allowed

rename          = "RENAME" SP mailbox SP mailbox
                  [SP partition] [SP rename_params]
                  ;; Use of INBOX as a destination gives
                  ;; a NO error

```



```
rename_params      = "(" rename_param *( SP rename_param ) ")"

rename_param_name = atom
                  ;; IMAPABNF tagged-ext-label?

rename_param       = rename_param_name SP rename_param_value

rename_param_value= astring / "(" astring *(SP astring) ")"
                  ;; As SELECT parameters in ANNOTATE.
                  ;; <<This should be updated to match
                  ;; IMAPABNF>>
```

6. IANA considerations

<<TBD>>

7. Acknowledgments

The author would like to thank Cyrus Daboo for the initial motivation for this document.

8. Normative References

[KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [RFC 2119](#), March 1997.

[ABNF] Crocker, Overell, "Augmented BNF for Syntax Specifications: ABNF", [RFC 2234](#), Internet Mail Consortium, Demon Internet Ltd, November 1997.

[IMAP4] Crispin, M., "Internet Message Access Protocol - Version 4rev1", [RFC 3501](#), University of Washington, March 2003.

[IMAPABNF] Melnikov, A., and C. Daboo, "Collected extensions to IMAP4 ABNF", work in progress, [draft-melnikov-imap-ext-abnf-XX.txt](#).

9. Author's Addresses

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

Email: Alexey.Melnikov@isode.com

URI: <http://www.melnikov.ca/>

10. Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

11. Full Copyright Statement

Copyright (C) The Internet Society (2005).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM 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.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

