

IMAP4 LIST Command Extensions

Status of this Document

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#). 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 an Proposed Standard for the Internet Community.

Discussion and suggestions for improvement are requested, and should be sent to ietf-imapext@imc.org. This document will expire before 31 August 2003. Distribution of this memo is unlimited.

1. Abstract

IMAP4 has two commands for listing mailboxes: LIST and LSUB. As we have added extensions that have required specialized lists (see [MboxRefer] for an example) we have had to expand the number of list commands, since each extension must add its function to both LIST and LSUB, and these commands are not, as they are defined, extensible. If we've needed the extensions to work together, we've had to add a set of commands to mix the different options, the set increasing in size with each new extension. This document describes an extension to the base LIST command that will allow these additions to be done with mutually compatible options to the LIST command, avoiding the exponential increase in specialized list commands.

2. 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 words "MUST", "MUST NOT", "SHOULD", "SHOULD NOT", and "MAY" are used in this document as specified in [RFC 2119](#) [Keywords].

3. Introduction

The extensions to the LIST command will be accomplished by amending the syntax to allow options to be specified. The list of options will replace the several commands that are currently used to mix and match the information requested. The new syntax is backward-compatible, with no ambiguity: if the first word after the command name begins with a parenthesis, the new syntax is being used; if it does not, it's in the original syntax.

By adding options to the LIST command, we are announcing the intent to phase out and eventually to deprecate the RLIST and RLSUB commands described in [MboxRefer]. We are also defining the mechanism to request extended mailbox information, such as is described in the now-expired "Child Mailbox Extension" draft [ChildMbox]. The base LSUB command is not deprecated by this extension; rather, this extension adds a way to obtain subscription information with more options, with those server implementations that support it. Clients that simply need a list of subscribed mailboxes, as provided by the LSUB command, SHOULD continue to use that command.

4. LIST Command Options

The LIST command syntax is extended by adding a parenthesized list of command options between the command name and the reference name (see the formal syntax in [section 6](#) for specific details). Command options will be defined in this document and in approved extension documents; each option will be enabled by a capability string (one capability may enable multiple options), and a client MUST NOT send an option for which the server has not advertised support. A server MAY ignore options it does not recognize.

This extension is identified by the capability string "LISTEXT", and support for it is a prerequisite for any future extensions that require specialized forms of the LIST command. Such extensions MUST refer to this document and MUST add their function through command options as described herein. This document also defines the "LIST-

SUBSCRIBED" capability string; see the "SUBSCRIBED" option below.

This extension also defines extensions to the LIST response, allowing a series of extended fields at the end, a parenthesized list of attribute/value pairs. Each attribute is a string, each value may be a string or a nested parenthesized list of the same attribute/value pairs. An example of this extended set might be

```
((("ACL" (("user" "fred")("access" "rwi")))("X-Sample" "text"))
```

See the formal grammar, below, for the full syntactic details.

The options defined in this specification are

SUBSCRIBED - causes the LIST command to list subscribed

mailboxes, rather than the actual mailboxes. This will often be a subset of the actual mailboxes. It's also possible for this list to contain the names of mailboxes that don't exist. In any case, the list MUST include exactly those mailbox names that match the selection criteria and are subscribed to. This option is intended to supplement the LSUB command, and support for it is optional -- a server that supports the SUBSCRIBED option indicates so through the LIST-SUBSCRIBED capability. Of particular note are the mailbox flags as returned by this option, compared with what is returned by LSUB. With the latter, the flags returned may not reflect the actual flag status on the mailbox, and the \NoSelect flag has a special meaning (it indicates that this mailbox is not, itself, subscribed, but that it has child mailboxes that are). With the SUBSCRIBED option described here, the flags are accurate and complete, and have no special meanings.

"LSUB" and "LIST (SUBSCRIBED)" are, thus, not the same thing, and some servers must do significant extra work to respond to "LIST (SUBSCRIBED)". Because of this, clients SHOULD continue to use "LSUB" unless they specifically want the additional information offered by "LIST (SUBSCRIBED)". At the same time, servers SHOULD support the LIST-SUBSCRIBED capability even if it entails extra work, because a client that wants the information will still obtain it by using LSUB followed by a series of LIST commands, so servers might as well make it easier.

This option defines a new mailbox flag, "\NonExistent", that indicates that a mailbox is subscribed to, but does not actually exist. The "\NonExistent" flag MUST be supported and MUST be accurately computed.

REMOTE - causes the LIST command to show remote mailboxes as well as local ones, as described in [MboxRefer]. This option

is intended to replace the RLIST command and, in conjunction with the SUBSCRIBED option, the RLSUB command. This option is only available on servers that also support [RFC-2193](#).

CHILDREN - Requests mailbox child information as originally proposed in [ChildMbox]. See [section 5](#), below, for details. Support for this is optional, but this option MUST be accepted by all servers (though it MAY be ignored).

The LISTEXT capability also defines a new mailbox flag, "\Placeholder", that indicates that the designated mailbox does not meet the selection criteria of the given LIST command, but that it has one or more child mailboxes that do [EDITORIAL NOTE: "might?"]. The LSUB command indicates this condition by using the "\NoSelect" flag, but the LIST (SUBSCRIBED) command MUST NOT do that, since "\NoSelect" retains its original meaning here. Further, the "\Placeholder" flag is more general, in that it can be used with any extended set of selection criteria.

5. The CHILDREN Option

The CHILDREN option implements the Child Mailbox Extension, originally proposed by Mike Gahrns and Raymond Cheng, of Microsoft Corporation. Most of the information in this section is taken directly from their original specification [ChildMbox]. The CHILDREN option is simply an indication that the client wants this information; a server MAY provide it even if the option is not specified, or MAY ignore the option entirely.

Many IMAP4 [IMAP4] clients present to the user a hierarchical view of the mailboxes that a user has access to. Rather than initially presenting to the user the entire mailbox hierarchy, it is often preferable to show to the user a collapsed outline list of the mailbox hierarchy (particularly if there is a large number of mailboxes). The user can then expand the collapsed outline hierarchy as needed. It is common to include within the collapsed hierarchy a visual clue (such as a '+'') to indicate that there are child mailboxes under a particular mailbox. When the visual clue is clicked the hierarchy list is expanded to show the child mailboxes. The Child Mailbox Extension provides a mechanism for a client to efficiently determine if a particular mailbox has children, without issuing a LIST '' * or a LIST '' % for each mailbox name.

The Child Mailbox Extension defines two new attributes that MAY be returned within a LIST response: \HasChildren and \HasNoChildren. While these attributes MAY be returned in response to any LIST command, the CHILDREN option is provided to indicate that the client

particularly wants this information. If the CHILDREN option is present, the server SHOULD return these attributes even if their computation is expensive.

\HasChildren - The presence of this attribute indicates that the mailbox has child mailboxes.

A server SHOULD NOT set this attribute if there are child mailboxes, and the user does not have permissions to access any of them. In this case, \HasNoChildren SHOULD be used.

In many cases, however, a server may not be able to efficiently compute whether a user has access to all child mailboxes. As such a client MUST be prepared to accept the \HasChildren attribute as a hint. That is, a mailbox MAY be flagged with the \HasChildren attribute, but no child mailboxes will appear in the LIST response.

\HasNoChildren - The presence of this attribute indicates that the mailbox has NO child mailboxes that are accessible to the currently authenticated user.

In some instances a server that supports the Child Mailbox Extension might not be able to determine whether a mailbox has children. For example it may have difficulty determining whether there are child mailboxes when LISTing mailboxes while operating in a particular namespace.

In these cases, a server MAY exclude both the \HasChildren and \HasNoChildren attributes in the LIST response. As such, a client can not make any assumptions about whether a mailbox has children based upon the absence of a single attribute. In particular, some servers may not be able to combine the SUBSCRIBED and CHILDREN options. Such servers MUST honour the SUBSCRIBED option, and they will simply ignore the CHILDREN option if both are requested.

It is an error for the server to return both a \HasChildren and a \HasNoChildren attribute in a LIST response.

Note: the \HasNoChildren attribute should not be confused with the IMAP4 [IMAP4] defined attribute \NoInferiors which indicates that no child mailboxes exist now and none can be created in the future.

6. Examples

The first example shows the complete local hierarchy that will be used for the other examples.

```
C: A01 LIST "" "*"
S: * LIST (\Marked \NoInferiors) "/" "inbox"
S: * LIST () "/" "Fruit"
S: * LIST () "/" "Fruit/Apple"
S: * LIST () "/" "Fruit/Banana"
S: * LIST () "/" "Tofu"
S: * LIST () "/" "Vegetable"
S: * LIST () "/" "Vegetable/Broccoli"
S: A01 OK done
```

In the next example, we'll see the subscribed mailboxes. This is similar, but not equivalent, to <LSUB "" ">. Note that the mailbox called "Fruit/Peach" is subscribed to, but does not actually exist (perhaps it was deleted while still subscribed). And the "Fruit" mailbox is not subscribed to, but it has two subscribed children.

```
C: A02 LIST (SUBSCRIBED) "" "*"
S: * LIST (\Marked \NoInferiors) "/" "inbox"
S: * LIST (\PlaceHolder) "/" "Fruit"
S: * LIST () "/" "Fruit/Banana"
S: * LIST (\NonExistent) "/" "Fruit/Peach"
S: A02 OK done
```

The next example shows the use of the CHILDREN option. The client, without having to list the second level of hierarchy, now knows which of the top-level mailboxes have sub-mailboxes (children) and which do not. Note that it's not necessary for the server to return the \HasNoChildren flag for the inbox, because the \NoInferiors flag already implies that, and has a stronger meaning.

```
C: A03 LIST (CHILDREN) "" "%"
S: * LIST (\Marked \NoInferiors) "/" "inbox"
S: * LIST (\HasChildren) "/" "Fruit"
S: * LIST (\HasNoChildren) "/" "Tofu"
S: * LIST (\HasChildren) "/" "Vegetable"
S: A03 OK done
```

In this example we see more mailboxes, which reside on another server to which we may obtain referrals. This is similar to the command <RLIST "" ">. We also see the mixing of two options. Note that in the case of the remote mailboxes, the server might or might not be able to include CHILDREN information; it includes it if it can, and omits it if it can't.

```
C: A04 LIST (REMOTE CHILDREN) "" "%"
S: * LIST (\Marked \NoInferiors) "/" "inbox"
S: * LIST (\HasChildren) "/" "Fruit"
S: * LIST (\HasNoChildren) "/" "Tofu"
S: * LIST (\HasChildren) "/" "Vegetable"
```

```

S: * LIST () "/" "Bread"
S: * LIST (\HasChildren) "/" "Meat"
S: A04 OK done

```

7. Formal Syntax

The following syntax specification uses the augmented Backus-Naur Form (BNF) as described in [ABNF]. Terms not defined here are taken from [IMAP4].

```

child-mbox-flag = "\HasChildren" / "\HasNoChildren"
                  ; flags for Child Mailbox Extension, at most one
                  ; possible per LIST response
list             = "LIST" [SP list-options] SP mailbox SP list-mailbox
list-options     = "(" [option *(SP option)] ")"
mailbox-list     = "(" [mbx-list-flags] ")" SP
                  (DQUOTE QUOTED-CHAR DQUOTE / nil) SP mailbox
                  [SP mbox-list-extended]
mbox-list-extended = "(" mbox-list-extended-item
                      *(SP mbox-list-extended-item) ")"
mbox-list-extended-item = "(" string SP (string / mbox-list-extended-
item) ")"
                      / mailbox-list-extended
mbox-list-oflag := child-mbox-flag / "\NonExistent" / "\Placeholder"
option           = "SUBSCRIBED" / "CHILDREN" / "REMOTE" /
                  option-extension
option-extension = atom

```

8. Security Considerations

This document describes syntactic changes to the specification of the IMAP4 commands LIST, LSUB, RLIST, and RLSUB, and the modified LIST command has the same security considerations as those commands. They are described in [IMAP4] and [MboxRefer].

The Child Mailbox Extension provides a client a more efficient means of determining whether a particular mailbox has children. If a mailbox has children, but the currently authenticated user does not have access to any of them, the server SHOULD respond with a

\HasNoChildren attribute. In many cases, however, a server may not be able to efficiently compute whether a user has access to all child mailboxes. If such a server responds with a \HasChildren attribute, when in fact the currently authenticated user does not have access to any child mailboxes, potentially more information is conveyed about the mailbox than intended. In most situations this will not be a security concern, because if information regarding whether a mailbox has children is considered sensitive, a user would not be granted access to that mailbox in the first place.

9. References

[IMAP4]; Crispin, M.; "Internet Message Access Protocol - Version 4rev1"; [RFC 2060](#); University of Washington; December 1996.
[MboxRefer]; Gahrns, M.; "IMAP4 Mailbox Referrals"; [RFC 2193](#); Microsoft Corporation; September 1997.
[ChildMbox]; Gahrns, M. & Cheng, R.; "IMAP4 Child Mailbox Extension"; [RFC 3328](#); Microsoft Corporation; July 2002.
[Keywords]; Bradner, S.; "Key words for use in RFCs to Indicate Requirement Levels"; [RFC 2119](#); Harvard University; March 1997.
[ABNF]; Crocker, D., and Overell, P. "Augmented BNF for Syntax Specifications: ABNF", [RFC 2234](#), November 1997.

10. Acknowledgements

Mike Gahrns and Raymond Cheng of Microsoft Corporation originally devised the Child Mailbox Extension and proposed it in 1997; the idea, as well as most of the text in [section 5](#), is theirs.

11. Author's Address

Barry Leiba

IBM T.J. Watson Research Center
30 Saw Mill River Road
Hawthorne, NY 10532
Phone: 1-914-784-7941
Email: leiba@watson.ibm.com