Network Working Group
Internet-Draft
Updates: 1939 (if approved)
Intended status: Standards Track
Expires: February 2, 2012

S. Lehmann
Strato AG

Aug 2011

The Post Office Protocol (POP3) LIST+ Extension
draft-lehmann-morg-pop3listplus-01

Abstract

The Post Office Protocol - Version 3 (POP3) LIST+ Extension allows a POP3 client to instruct a POP3 server to send additional information with the LIST command response. It also can instruct a POP3 server to send only information about newly arrived messages, when polling for new messages. This will help save message round trips and network bandwidth, especially on large mailboxes containing several messages. On public POP3 servers, serving thousands of mailboxes, and being polled frequently, LIST+ will help reduce server load as well as network bandwith.

Note

This document defines an extension to POP3, and updates RFC 1939.

Technical comments are solicited and should be addressed to Message Organizations Working Group's mailing list at morg@ietf.org and/or to the author. You can subscribe to that list at https://www.ietf.org/mailman/listinfo/morg.

Editorial comments should be addressed directly to the author.

Status of this Memo

This Internet-Draft will expire on February 2, 2012.

Copyright Notice

Table of Contents

1. Introduction

1.1. Overview

Although POP3 is intended as a download-and-delete protocol, clients can decide to leave messages on the server. If this happens on a mailbox having "unlimited" mail space, such a mailbox will grow continously in size, so mailboxes having thousands of mails are not uncommon. The POP3 protocol has no means to poll for only new arrived messages. Observations on a public POP3 server have shown several POP3 clients using the LIST command, and, immediately followed, the UIDL command to get the Message Number, Message Size and Unique-ID, of all messages of a mailbox, to see if new messages have arrived. Both commands, LIST and UIDL, always respond with a list of all messages. On large mailboxes, this will waste network bandwith, and will prolong the duration a mailbox is hold in locked state. This situation is even stressed by the fact that POP3 clients will usually poll a mailbox for new messages periodically in short intervals.

The LIST command, on the server, has to scan through all messages of a mailbox to provide the Message Number and the Message Size. The subsequent UIDL command has to scan again through all messages of a mailbox to provide the Message Number and the Unique-ID. The Message Number sent with the UIDL command, in this scenario, is already known to the client, and therefore, is a redundant information (as in example shown in Appendix A.1). All the lines with Message Numbers the client already knows, are of no interest by client, and therefore, sent by server in vain. Using the LIST+ Extension, a server has to scan through the mailbox only for a single time, and can send the Message Number, Message Size and Unique-ID in a more compact form to the client (as shown in Appendix A.2.1).

Another major benefit of the LIST+ Extension is that the client can instruct the server to send only information about newly arrived messages. This avoids sending of information by server, which the client already knows.

The LIST+ Extension also offers the possibility to fetch other useful information with the LIST command quasi "on the fly", like the age of a message. This may be helpful for clients to decide if a message has to be deleted automatically due to clients' message deleting policies. There is no need to read the mail headers to make this decision in this case.

Although it would be better to access large mailboxes by IMAP, there might be reasons for customers to use POP3.

1.2. 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].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

In this document, the term "client" refers to a host making use of the POP3 service, while the term "server" refers to a host which offers the POP3 service.

In examples, "C:" and "S:" indicate lines sent by the client and server respectively.

In this document, the term "scan listing" refer to the output of a POP3 LIST command, containing one line for every message in a mailbox, as described in [RFC1939], Section 5.

In this document, the terms "extended scan listing" and "full scan listing" refer to a "POP3 scan listing", which can hold additional information, controlled by the means of the POP3 LIST+ command, as described in this document.

2. The LIST+ Capability

This section defines the LIST+ Capability according to Section 9 of [RFC2449].

CAPA tag:
   LIST+

Arguments:
   Supported LIST+ Flags

Added commands:
   none

Standard commands affected:
   LIST

Announces states / possible differences:
   both / no

Commands valid in states:
n/a

Specification reference:
This document and [RFC1939]

Discussion:
The LIST+ capability enables the use of the indicated LIST+ Flags as additional parameters of the LIST command.

3. The LIST Command

This section updates the syntax of the LIST command as well as the definition of the scan listing, both as defined in Section 5 of [RFC1939].

Formal syntax: LIST [msg] [flaglist]

The flaglist is a list of LIST+ Flags, separated by a single space. A LIST+ Flag always starts with "+" (it means "add the following information"), followed by the name of the flag. A flag name MUST start with a letter ("A-Z" or "a-z"), and MUST NOT be longer than 20 characters.

A server, supporting LIST+, has to decide whether the first argument of the LIST command (if any) is a message number or a LIST+ Flag. If an argument starts with "+", and its second character is a letter, this parameter is to be considered as a LIST+ flag.

If the server receives a LIST+ flag which is not supported, the server MUST issue a "-ERR" response.

After the server has accepted a flaglist, it sends an extended scan listing for each requested message back to the client. The extended scan listing consists of:

- The message-number of the message, followed by a single space

- The exact size of the message in octets, followed by a single space

- The value(s) as requested by the flaglist, separated each by a single space, in the same order as given in the flaglist. A value MUST consist of only ASCII characters in the range of 33-126 (hex 21-7E). Otherwise, it is RECOMMENDED to use BASE64 encoding according to [RFC4648].

See Appendix B for a formal syntax using ABNF.

4. Initial Set of LIST+ Flags

4.1. The +AGE Flag

The +AGE Flag requests to add the age of a message to the extended scan listing. The age is expressed in units of whole days since the message was delivered into the mailbox.

A message delivered yesterday (regardless of the time) has an age of one. A message delivered today has an age of zero. The value is presented as a decimal integer.

Servers which are unable to determine the age of a message in this way MUST NOT advertise the +AGE Flag in the LIST+ capability list.

The +AGE Flag has no Parameter.

4.2. The +ID Flag

4.2.1. Purpose

The +ID Flag instructs the server to

- suppress all scan lines the server has already issued (except the last one),
- send an ID-Identifier as first item in the "OK" response line.

The +ID Flag, when sent by client, always has a parameter. This parameter can be:

- The ID-Identifier as received from server in a response of a previous LIST+ command.
- An empty string.

4.2.2. The ID-Identifier

The ID-Identifier, if not empty, MUST consist of characters all in the range of 33-126 (hex 21-7E). Its length should be kept as short as possible, but MUST NOT exceed 255 bytes. The ID-Identifier is generated by the server. The server has to put all information into the ID-Identifier which it needs to find out the last issued message of a mailbox. A newly created ID-Indentifier MUST be unique with respect to former created ID-Identifiers.

The server stores the last issued ID-Indentifier in the meta-data store of a mailbox. A volatile memory MAY be used. It is not allowed to store more than one ID-Identifier per mailbox. The server MAY delete a stored ID-Identifier at any time.

The client retains a received ID-Identifier, and MAY send it as a +ID Flag parameter in the very next LIST+ command. After the client has sent a LIST command or LIST+ command (with or without +ID Flag), a stored ID-Identifier MUST be deleted by client.

4.2.3. Actions at server’s side

The server deletes one or more messages from the mailbox:
    If there is a stored ID-Identifier, delete it.

The server appends one or more messages to the mailbox:
    If there is a stored ID-Identifier, leave it intact.

The server receives a LIST command or LIST+ command without +ID Flag:
    Send a full scan listing. If there is a stored ID-Identifier, leave it intact. Send no ID-Identifier to the client.

The server receives a LIST+ command with an empty ID-Identifier:
    Send a full scan listing. If there is a stored ID-Identifier, leave it intact and sent it to the client. If there is no stored ID-Identifier, create one and send it to the client.

The server receives a LIST+ command with a non-empty ID-Identifier equal to the stored one:
    If the mailbox ist empty: Send an empty scan listing, leave the ID-Identifier intact and send it to the client. If the message, as referenced by the ID-identifier, is still the last message in the mailbox: Send a partial scan listing which constist of only the last message in the mailbox, leave the ID-Identifier intact and send it to the client. If the message, as referenced by the ID-Identifier, is not the last message in the mailbox: Send a partial scan listing which constist of all messages with message numbers above the message number as referenced by the ID-Identifier, create and store a new ID-Identifier and send it to the client.

The server receives a LIST+ command with a non-empty ID-Identifier not equal to the stored one:
    Send a full scan listing. If there is a stored ID-Identifier, leave it intact and send it to the client. If there is no stored ID-Identifier, create one and send it to the client.

4.2.4. Actions at client’s side

The client receives an empty scan listing:
Consider the mailbox as empty. Store the new ID-Identifier.

The client receives a scan listing with first message number equal to one:
Handle the scan listing as a full scan listing. Store the new ID-Identifier.

The client receives a scan listing with first message number greater than one:
Handle the scan listing as a partial scan listing. Store the new ID-Identifier. Note: The client could already know the message as indicated in the very first scan line. In this case, the clients MAY skip processing of this scan line.

4.2.5. Examples

The client has just started up. For this reason, it has no ID-Identifier yet, and sends an empty ID-Identifier to request one. The servers issues all messages and an ID-Identifier "Az-Df˜701", which is retained by client:

```
S: +OK POP3 server ready
C: CAPA
S: +OK Capability list follows:
S: LIST+ +AGE +ID +UIDL
S: .
C: LIST +ID= +UIDL
S: +OK Az-Df˜701 10299 messages, listing follows:
S: 1 4536 7abcb6f4da22080e121f2824aef2be9a
S: 2 4422 577151e65ea4e62bb9b1c5830a818fe7
S: 3 1134 6374234adfg58cda7b4ccdf34bccacc6
S: ... (10295 more lines)
S: 10299 3828 2fe542ac6d1c29bb6a5161558f527e2e
S: .
C: QUIT
S: +OK Closing connection
```

Client connecting the second time when there are no changes. The server issues the ID-Identifier "Az-Df˜701" again, which is retained by client, and sends only the the last message in a partial scan listing:

```
   S: +OK POP3 server ready
   C: CAPA
   S: +OK Capability list follows:
   S: LIST+ +AGE +ID +UIDL
   S: .
   C: LIST +ID=Az-Df˜701 +UIDL
   S: +OK Az-Df˜701 No changes
   S: 10299 3828 2fe542ac6d1c29bb6a5161558f527e2e
   S: .
   C: QUIT
   S: +OK Closing connection
```

Client connecting the 3rd time when there are two new messages. The server issues the ID-Identifier "Az-Df˜702", which is retained by client. Only the two new messages are sent in a partial scan listing:

```
   S: +OK POP3 server ready
   C: CAPA
   S: +OK Capability list follows:
   S: LIST+ +AGE +ID +UIDL
   S: .
   C: LIST +ID=Az-Df˜701 +UIDL
   S: +OK Az-Df˜702 2 more messages, listing follows:
   S: 10300 258 4c2674df036aba49035134701bc47401
   S: 10301 496 764fd5a52324c683c6b33a9c4c50abe2
   S: .
   C: QUIT
   S: +OK Closing connection
```

Client connecting the 4th time when message #2 was deleted. The server issues a complete scan listing, because of message deletetion. The new ID-Identifier is "Az-Df˜703".

```
   S: +OK POP3 server ready
   C: CAPA
   S: +OK Capability list follows:
   S: LIST+ +AGE +ID +UIDL
   S: .
   C: LIST +ID=Az-Df˜702 +UIDL
   S: +OK Az-Df˜703 10300 messages, listing follows:
   S: 1 4536 7abcb6f4da22080e121f2824aef2be9a
   S: 2 1134 6374234adfg58cda7b4ccdf34bccacc6
   S: ... (10297 more lines)
   S: 10300 496 764fd5a52324c683c6b33a9c4c50abe2
   S: .
   C: QUIT
   S: +OK Closing connection
```

4.3. The +UIDL Flag

The +UIDL Flag requests to add the Unique-ID of a message to the extended scan listing. The Unique-ID for a message is the same value as it would be sent by the UIDL command for this message. See [RFC1939], Section 7, "UIDL Command".

Servers not supporting the UIDL command MUST NOT advertise the +UIDL Flag in the LIST+ capability list.

The +UIDL Flag has no Parameter.

5. Parameter and Response Length

The Parameter and Response Lengths, as defined in Section 4 of [RFC2449], are not affected by the LIST+ Extension Flags as defined in this document.

The already defined Parameter and Response Lengths are sufficient to hold complete LIST+ command and response lines as defined in this document, even if all possible flags are given together.

6. Future Extensions to LIST+

The LIST+ Extension itself is expandable. New LIST+ flags may be defined by future documents. If such a document defines a LIST+ flag whose value contains characters outside the range as defined in Section 3, it MUST declare how to encode the value, so it fulfills the requirements of this document.

Future documents defining new LIST+ flags MUST contain a statement whether the Parameter and Response Length, as defined in Section 4 of [RFC2449], are sufficient to hold lines with all possible flags/ values, or not. If not, new values for parameter and response length MUST be defined.

7. Security Considerations

The LIST+ Extension, if not proper implemented by client or server, could prevent mails to be fetched from server. Firewalls or other network devices, when monitoring POP3 sessions, could be worried by the protocol changes made by LIST+.

8. IANA Considerations

8.1. ’LIST+ Flags’ Registry

IANA is asked to create and maintain the ’LIST+ Flags’ registry following the guidelines below.

The registry consists of two values: LIST+ flag and Reference. They are described below.

1) LIST+ flag - the text string, whose syntax is defined in Section 3.

2) Reference - the reference to the document that described the LIST+ flag.

Initial values are given below; new assignments are to be made following the ’IETF Consensus’ policies [RFC5226].

LIST+ Flag	Reference
+UIDL	this document
+AGE	this document
+ID	this document

Table 1: Initial IANA Values

8.2. LIST+ Registration in ’POP3 Capabilities’ Registry

> IANA is asked a new keyword "LIST+" in the "POP3 Capabilities" registry [RFC1939] using the template in Section Section 2 of this document.

9. Conclusions

All mail program/server developers are encouraged to implement LIST+.

All carriers of POP3 servers holding mailboxes of many, frequently polling clients are encouraged to support LIST+.

10. Normative References

[RFC1939] Myers, J. and M. Rose, "Post Office Protocol - Version 3", STD 53, RFC 1939, May 1996.

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

[RFC2449] Gellens, R., Newman, C., and L. Lundblade, "POP3 Extension Mechanism", RFC 2449, November 1998.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, October 2006.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

Appendix A. Message flow examples

A.1. Typical message flow without LIST+ Extension

```
   S: +OK POP3 server ready
   C: CAPA
   S: +OK Capability list follows:
   S: UIDL
   S: .
   C: LIST
   S: +OK 3 messages, listing follows:
   S: 1 4536
   S: 2 4422
   S: ... (10296 more lines)
   S: 10299 3828
   S: .
   C: UIDL
   S: +OK 10299 messages, listing follows:
   S: 1 7abcb6f4da22080e121f2824aef2be9a
   S: 2 577151e65ea4e62bb9b1c5830a818fe7
   S: ... (10296 more lines)
   S: 10299 2fe542ac6d1c29bb6a5161558f527e2e
   S: .
   C: QUIT
   S: +OK Closing connection
```

A.2. Message Flow using the LIST+ Extension

A.2.1. LIST+ for all messages

```
   S: +OK POP3 server ready
   C: CAPA
   S: +OK Capability list follows:
   S: UIDL
   S: LIST+ +UIDL +AGE
   S: .
   C: LIST +UIDL
   S: +OK 10299 messages, listing follows:
   S: 1 4536 7abcb6f4da22080e121f2824aef2be9a
   S: 2 4422 577151e65ea4e62bb9b1c5830a818fe7
   S: ... (10296 more lines)
   S: 10299 3828 2fe542ac6d1c29bb6a5161558f527e2e
   S: .
   C: QUIT
   S: +OK Closing connection
```

A.2.2. LIST+ for a single message

```
   S: +OK POP3 server ready
   C: CAPA
   S: +OK Capability list follows:
   S: UIDL
   S: LIST+ +UIDL +AGE
   S: .
   C: LIST 2 +UIDL
   S: +OK 2 4422 577151e65ea4e62bb9b1c5830a818fe7
   C: QUIT
   S: +OK Closing connection
```

A.2.3. LIST+ for all messages using all possible LIST+ Flags, one new message arrived

```
   S: +OK POP3 server ready
   C: CAPA
   S: +OK Capability list follows:
   S: UIDL
   S: LIST+ +UIDL +AGE
   S: .
   C: LIST +UIDL +AGE +ID=Az-Df˜701
   S: +OK Az-Df˜702 1 new message, listing follows:
   S: 10300 4536 ead6763cdc00080f36172429ac13c1cb 0
   S: .
   C: QUIT
   S: +OK Closing connection
```

Appendix B. Augmented BNF

This section defines the LIST+ syntax using Augmented BNF (see

[RFC5234] for details). The following rules are for clarification only, and are not intended to supersede the definition of the LIST command, as defined in [RFC1939].

```
listplus-req      = "LIST" [SP msg-num] [SP listplus-flaglist] CRLF
listplus-rsp      = listplus-rsp-ok / listplus-rsp-err
listplus-rsp-ok   = "+OK" [SP listplus-id] [SP *comment] CRLF
                        *listplus-scanline
                        "." CRLF
listplus-rsp-err  = "-ERR" [SP *comment] CRLF
listplus-scanline = msg-num SP msg-size [*(SP listplus-value)] CRLF
listplus-flaglist = listplus-flag [*(SP listplus-flag)]
listplus-flag     = "+" ALPHA [*listplus-char] ["=" listplus-value]
listplus-value    = 1*listplus-char
listplus-id       = 1*255listplus-char
listplus-char     = %x21-7E
msg-num           = 1*DIGIT
msg-size          = 1*DIGIT
comment           = *(WSP / VCHAR)
```

Appendix C. Acknowledgements

Timo Sirainen contibuted an invaluable suggestion with the +ID flag. Also, thanks to Mykyta Yevstifeyev, and many others.

Author’s Address

Steffen Lehmann
Strato AG
Pascalstr. 10
Berlin D-10587
GERMANY

Email: lehmann@strato-rz.de
URI: http://www.strato.de/