

IMAP4 COMMAND+ Extension

Status of this Memo

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/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

Abstract

The COMMAND+ extension to the Internet Message Access Protocol [[IMAP4](#)] defines a formal syntax for extending [[IMAP4](#)] commands. This includes adding requirements on future extensions to the [[IMAP4](#)] syntax that may add new commands, so that any command in an [[IMAP4](#)] session, be it part of the base-specification, or a current or future extension, can be extended in an arbitrary manner, without conflicts in syntax between current or future extensions.

Copyright Notice

Copyright (C) The Internet Society (1999). All Rights Reserved.

1. Introduction and Overview

The COMMAND+ extension is present in any IMAP4 implementation which returns "COMMAND+" as one of the supported capabilities to the CAPABILITY command.

The COMMAND+ extension specifies a new syntax for all IMAP4 commands.

The design goal for COMMAND+ is to ensure an easy way to extend IMAP4 commands with "options", defined in other extension documents. By following the design rules laid out in this document, authors of future extensions can ensure that their extension syntax will allow other extensions to coexist.

The goal of this extension is to remove the need to define separate commands that modify the behaviour of an existing IMAP4 command, and therefore to cut down on the number of possibilities where multiple options for a single IMAP4 command are required. The aim is to provide a framework for adding 'options' to any IMAP4 command in an extensible manner, that will work with both current and future extensions.

2. Conventions Used in This Document

The key words "REQUIRED", "MUST", "MUST NOT", "SHOULD", "SHOULD NOT", and "MAY" in this document are to be interpreted as described in "Key words for use in RFCs to Indicate Requirement Levels" [[KEYWORDS](#)].

Formal syntax is defined using the augmented Backus-Naur Form (BNF) notation as specified in [[RFC-822](#)] as modified by [[IMAP4](#)].

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

This document uses the following terms to describe an IMAP4 command sent by the client to the server.

command

refers to the [[IMAP4](#)] command sent by a client to a server.

Note this can apply to commands introduced by extensions, not just commands in the base specification [[IMAP4](#)].

tag

refers to the [[IMAP4](#)] tag at the start of every command line.

command descriptor

refers to the "base" unmodified command syntax following the tag as defined in the "base" document for the corresponding command.

Daboo, Leiba

Expires October 1999

[Page 2] Intern

command options

a set of options, as defined in a separate extension document, that extends a command.

option arguments

a set of arguments that can be applied to an option to modify the behaviour the option has on the corresponding command.

option data

a set of data "objects" added to the end of a command when a command option needs to include large volumes of information in an option argument.

[3.](#) Modification to IMAP command syntax

When the COMMAND+ extension is in effect, any IMAP4 command MAY accept command options that modify the behaviour of that command.

To specify command options, the client MUST prefix the command descriptor with the command option list. The options themselves are defined in separate extensions that MUST be advertised as one of the supported capabilities of the CAPABILITY command. Each option MAY also include option arguments that allow the command option to be further extended if required.

Example:

```
C: A001 CAPABILITY
S: * CAPABILITY IMAP4rev1 COMMAND+ LISTCHILDREN LISTREFERRAL
S: A001 OK
...
C: A006 (LISTCHILDREN () LISTREFERRAL (ALL)) LIST "" "%"
S: ...
S: A006 OK
```

In the above example, the "LISTCHILDREN" and "LISTREFERRAL" command options are added by the client to the LIST command. These command options modify the server behaviour for the LIST command based on

the corresponding extension document defining these extensions. The "LISTREFERRAL" command option includes the "ALL" option argument which modifies the behaviour of the command option in some particular manner.

A command option may define that additional option data is appended to the end of the command descriptor. This may be useful when the command option requires an argument whose size may be large, and the processing of which is better deferred until after the base command has been processed by the server.

Example:

```
C: A006 (ADDNOTES ()) CREATE INBOX.test ({19}
S: + Ready for literal data
C: Here are some notes)
```

Daboo, Leiba

Expires October 1999

[Page 3] Intern

```
S: A006 OK
```

Extensions that allow this type of behaviour with the [\[IMAP4\]](#) SEARCH command MUST require that the search keys are explicitly surrounded by parenthesis, which are normally implicit to the SEARCH command.

Example:

```
C: A007 (MORE ()) SEARCH (SUBJECT FROM) ("extra data")
```

If multiple command options add option data to the end of a command descriptor, then the option data MUST appear in the same order as the command options appear in the command option list.

Example:

```
C: A007 (X () Y () Z ()) SEARCH (SUBJECT) ("X-data") ("Z-data")
```

In order to support this extension the following must be adhered to:

- * A server that advertises a capability that has a command option associated with it MUST be prepared to accept that command option.
- * A server MUST respond with a tagged BAD to any command that contains a command option for which the server did not advertise support.
- * A client MUST NOT send a command option for which the server did not advertise support.
- * A client MUST NOT send an empty command option list.
- * Each command option MUST include a parenthesised option

argument list that MAY be empty.

* A command option MAY define that additional option data be added after the command descriptor.

* Additional option data at the end of a command descriptor MUST appear in the same order as the corresponding command options in the command option list.

* Future extensions MUST NOT define a command descriptor that begins with the "(" character.

[4.](#) Formal Syntax

The following syntax specification uses the augmented Backus-Naur Form (BNF) notation as specified in [[RFC-822](#)] as modified by [[IMAP4](#)]. Non-terminals referenced but not defined below are as defined by [[IMAP4](#)].

Daboo, Leiba

Expires October 1999

[Page 4] Intern

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.

```
command_ex      ::= tag SPACE
                  ["(" command_opt [*(SPACE command_opt)] ")" SPACE]
                  (command_any / command_auth / command_nonauth /
                   command_select / command_extension)
                  [*(SPACE option_data)] CRLF
```

```
command_extension ::=      ;; command defined by an extension specification
```

```
command_opt      ::= option_name SPACE
                  "(" [option_arg *(SPACE option_arg)] ")"
```

```
option_name      ::= 1*ATOM_CHAR
```

```
                ;; defined by extension specification
```

```
option_arg       ::= astring / nstring / number / snumber /
                  "(" option_arg *(SPACE option_arg) ")"
```

```
                ;; defined by extension specification
```

```
option_data      ::= "(" option_arg *(SPACE option_arg) ")"
```

;; defined by extension specification

snumber ::= ["+" / "-"] 1*digit

;; signed 32-bit integer

;; (-2147483648 <= n <= 2147483647)

5. Security Considerations

Command options that modify any command relating to security (e.g. authentication or security layer) MUST take security considerations into account when defined.

6. References

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

[IMAP4] Crispin, M., "Internet Message Access Protocol Version 4rev1", [RFC 2060](#), University of Washington, December 1996.

[RFC-822] Crocker, D., "Standard for the Format of ARPA Internet Text Messages", STD 11, [RFC 822](#), University of Delaware, August 1982.

Daboo, Leiba

Expires October 1999

[Page 5] Intern

7. Acknowledgments

Mark Pustilnik and Bart Schaefer have made important contributions to the creation and refinement of the ideas expressed in this document.

8. Authors' Addresses

Cyrus Daboo
Cyrusoft International, Inc.
Suite 780, 5001 Baum Blvd.
Pittsburgh, PA 15213

Phone: (412)605-0499
Email: daboo@cyrusoft.com

Barry Leiba
IBM T.J. Watson Research Center
30 Saw Mill River Road
Hawthorne, NY 10532

Email: leiba@watson.ibm.com