

IMAP4 IDLE command

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

1. Abstract

The Internet Message Access Protocol [IMAP4] requires a client to poll the server for changes to the selected mailbox (new mail, deletions). It's often more desirable to have the server transmit updates to the client in real time. This allows a user to see new mail immediately. It also helps some real-time applications based on IMAP, which might otherwise need to poll extremely often (such as every few seconds). (While the spec actually does allow a server to push EXISTS responses asynchronously, a client can't expect this behaviour and must poll.)

This document specifies the syntax of an IDLE command, which will allow a client to tell the server that it's ready to accept such real-time updates.

2. Conventions Used in this Document

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

The key words "MUST", "MUST NOT", "SHOULD", "SHOULD NOT", and "MAY" in this document are to be interpreted as described in RFC 2060 [IMAP4].

3. Specification

IDLE Command

Arguments: none

Responses: continuation data will be requested; the client sends the continuation data "DONE" to end the command

Result: OK - IDLE completed after client sent "DONE"
 NO - failure: the server will not allow the IDLE
 command at this time
 BAD - command unknown or arguments invalid

The IDLE command may be used with any IMAP4 server implementation that returns "IDLE" as one of the supported capabilities to the CAPABILITY command. If the server does not advertise the IDLE capability, the client MUST NOT use the IDLE command and must poll for mailbox updates. In particular, the client MUST continue to be able to accept unsolicited untagged responses to ANY command, as specified in the base IMAP specification.

The IDLE command is sent from the client to the server when the client is ready to accept unsolicited mailbox update messages. The server requests a response to the IDLE command using the continuation ("+") response. The IDLE command remains active until the client responds to the continuation, and as long as an IDLE command is active, the server is now free to send untagged EXISTS, EXPUNGE, and other messages at any time.

The IDLE command is terminated by the receipt of a "DONE" continuation from the client; such response satisfies the server's continuation request. At that point, the server MAY send any remaining queued untagged responses and then MUST immediately send the tagged response to the IDLE command and prepare to process other commands. As in the base specification, the processing of any new command may cause the sending of unsolicited untagged responses, subject to the ambiguity limitations. The client MUST NOT send a command while the server is waiting for the DONE, since the server will not be able to distinguish a command from a continuation.

The server MAY consider a client inactive if it has an IDLE command running, and if such a server has an inactivity timeout it MAY log the client off implicitly at the end of its timeout period. Because of that, clients using IDLE are advised to terminate the IDLE and re-issue it at least every 29 minutes to avoid being logged off. This still allows a client to receive immediate mailbox updates even though it need only "poll" at half hour intervals.

```

Example:  C: A001 SELECT INBOX
          S: * FLAGS (Deleted Seen)
          S: * 3 EXISTS
          S: * 0 RECENT
          S: * OK [UIDVALIDITY 1]
          S: A001 OK SELECT completed
          C: A002 IDLE
          S: + idling
          ..time passes; new mail arrives...
          S: * 4 EXISTS
          C: DONE
          S: A002 OK IDLE terminated
          ..another client expunges message 2 now...
          C: A003 FETCH 4 ALL
          S: * 4 FETCH (...)
          S: A003 OK FETCH completed
          C: A004 IDLE
          S: * 2 EXPUNGE
          S: * 3 EXISTS
          S: + idling
          ..time passes; another client expunges message 3...
          S: * 3 EXPUNGE
          S: * 2 EXISTS
          ..time passes; new mail arrives...
          S: * 3 EXISTS
          C: DONE
          S: A004 OK IDLE terminated
          C: A005 FETCH 3 ALL
          S: * 3 FETCH (...)
          S: A005 OK FETCH completed
          C: A006 IDLE

```

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].

```

command_auth ::= append / create / delete / examine / list / lsub /
               rename / select / status / subscribe / unsubscribe
               / idle
               ;; Valid only in Authenticated or Selected state

idle ::= "IDLE" CRLF "DONE"

```

5. References

[IMAP4] Crispin, M., "Internet Message Access Protocol - Version 4rev1", RFC 2060, December 1996.

6. Security Considerations

There are no known security issues with this extension.

7. Author's Address

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

Email: leiba@watson.ibm.com

