Network Working Group   Arnt Gulbrandsen
Internet-Draft   Oryx Mail Systems GmbH
Intended Status: Proposed Standard   December 15, 2008

                      IMAP Response Codes
           draft-gulbrandsen-imap-response-codes-07.txt

Status of this Memo

   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.

   This Internet-Draft expires in June 2009.

Abstract

   IMAP responses consist of a response type (OK, NO, BAD), an optional
   machine-readable response code and a human-readable text.

   This document collects and documents a variety of machine-readable
   response codes, for better interoperation and error reporting.

1. **Conventions Used in This Document**

   Formal syntax is defined by [RFC5234] as modified by [RFC3501].

   Example lines prefaced by "C:" are sent by the client and ones
   prefaced by "S:" by the server. "[...]" means elision.

2. **Introduction**

   [RFC3501] section 7.1 defines a number of response codes which can
   help tell an IMAP client why a command failed. However, experience
   has shown that more codes are useful. For example, it is useful for a
   client to know that an authentication attempt failed because of a
   server problem as opposed to a password problem.

   Currently many IMAP servers use English-language human-readable text
   to describe these errors, and a few IMAP clients attempt to translate
   this text into the user's language.

   This document names a variety of errors as response codes. It is
   based on errors checked and reported in some IMAP server
   implementations, and on needs in some IMAP clients.

   This document doesn't require any servers to test for these errors,
   or any clients to test for these names. It only names errors for
   better reporting and handling.

   [RFC Editor: Please remove this paragraph.] In general, this document
   aims to do that which is widely considered good, and nothing more.
   Several controversial and/or complex features were discussed, but
   just listing the simple and desirable response codes is enough for
   one document.

3. **Response Codes**

   This section defines all the new response codes. Each definition is
   followed by one or more examples.

UNAVAILABLE Temporary failure because a subsystem is down. For
            example, an IMAP server which uses an LDAP or Radius
            server for authentication might use this when the
            LDAP/Radius server is down.

            C: a LOGIN "fred" "foo"
            S: a NO [UNAVAILABLE] User's backend down for maintenance

AUTHENTICATIONFAILED Authentication failed for some reason which the
            server is not willing to elaborate. Typically this
            includes "unknown user" and "bad password".

            This is the same as not sending any response code, except
            that when a client sees AUTHENTICATIONFAILED, it knows
            that the problem wasn't e.g. UNAVAILABLE, so there's no
            point in trying the same login/password again later.

            C: b LOGIN "fred" "foo"
            S: b NO [AUTHENTICATIONFAILED] Authentication failed

AUTHORIZATIONFAILED Authentication succeeded, but authorization
            failed. This is only applicable when the authentication
            and authorization identities are different.

            C: c AUTHENTICATE PLAIN
            [...]
            S: c NO [AUTHORIZATIONFAILED] No such auth-ID

EXPIRED   Authentication succeeded or the server didn't have the
            necessary data any more, but access is no longer
            permitted using that passphrase. The client or user
            should get a new passphrase.

            C: d login "fred" "foo"
            S: d NO [EXPIRED] That password isn't valid any more

PRIVACYREQUIRED The operation is not permitted due to a lack of
            privacy. If TLS is not in use, the client could try
            STARTTLS (see [RFC3501] section 6.2.1) and then repeat
            the operation.

            C: d login "fred" "foo"
            S: d NO [PRIVACYREQUIRED] Connection offers no privacy

            C: d select inbox
            S: d NO [PRIVACYREQUIRED] Connection offers no privacy

CONTACTADMIN The user should contact the system administrator or
            support desk.

                    C: e login "fred" "foo"
                    S: e OK [CONTACTADMIN]

NOPERM   The access control system (e.g. ACL, see [RFC4314]) does
            not permit this user to carry out an operation, such as
            selecting or creating a mailbox.

                    C: f select "/archive/projects/experiment-iv"
                    S: f NO [NOPERM] Access denied

INUSE   An operation has not been carried out because it involves
            sawing off a branch someone else is sitting on. Someone
            else may be holding an exclusive lock needed for this
            operation, or it may involve deleting a resource someone
            else is using, typically a mailbox.

            The operation may succeed if the client tries again
            later.

                    C: g delete "/archive/projects/experiment-iv"
                    S: g NO [INUSE] Mailbox in use

EXPUNGEISSUED Someone else has issued an EXPUNGE for the same
            mailbox. The client may want to issue NOOP soon.
            [RFC2180] discusses this subject in depth.

                    C: h search from fred@example.com
                    S: * SEARCH 1 2 3 5 8 13 21 42
                    S: h OK [EXPUNGEISSUED] Search completed

CORRUPTION The server discovered that some relevant data (e.g. the
            mailbox) are corrupt. This response code does not include
            any information about what's corrupt, but the server can
            write that to its logfiles.

                    C: i select "/archive/projects/experiment-iv"
                    S: i NO [CORRUPTION] Cannot open mailbox

SERVERBUG   The server encountered a bug in itself or violated one of
            its own invariants.

                    C: j select "/archive/projects/experiment-iv"
                    S: j NO [SERVERBUG] This should not happen

    CLIENTBUG   The server has detected a client bug. This can accompany
                all of OK, NO and BAD, depending on what the client bug
                is.

                C: k1 select "/archive/projects/experiment-iv"
                [...]
                S: k1 OK [READ-ONLY] Done
                C: k2 status "/archive/projects/experiment-iv" (messages)
                [...]
                S: k2 OK [CLIENTBUG] Done

    CANNOT   The operation violates some invariant of the server and
                can never succeed.

                C: l create "///////"
                S: l NO [CANNOT] Adjacent slashes is not supported

    LIMIT   The operation ran up against an implementation limit of
                some kind, such as the number of flags on a single
                message or number of flags used in a mailbox.

                C: m STORE 42 FLAGS f1 f2 f3 f4 f5 ... f250
                S: m NO [LIMIT] At most 32 flags in one mailbox supported

    OVERQUOTA   The user is or would be over quota after the operation.
                (The user may or may not be over quota already.)

                Note that if the server sends OVERQUOTA but doesn't
                support the IMAP QUOTA extension defined by [RFC2087],
                then there is a quota, but the client cannot find out
                what the quota is.

                C: n1 uid copy 1:* oldmail
                S: n1 NO [OVERQUOTA] Sorry

                C: n2 uid copy 1:* oldmail
                S: n2 OK [OVERQUOTA] You are now over your soft quota

    ALREADYEXISTS The operation attempts to create something which
                already exists, such as when the CREATE or RENAME
                directories attempt to create a mailbox and there is one
                of that name.

                C: o RENAME this that
                S: o NO [ALREADYEXISTS] Mailbox "that" already exists

NONEXISTENT The operation attempts to delete something which does not
            exist. Similar to ALREADYEXISTS.

                C: p RENAME this that
                S: p NO [NONEXISTENT] No such mailbox

4. Formal Syntax

The following syntax specification uses the Augmented Backus-Naur
Form (ABNF) notation as specified in [RFC5234]. [RFC3501] defines the
non-terminal "resp-text-code".

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.

```
    resp-text-code =/ "UNAVAILABLE" / "AUTHENTICATIONFAILED" /
                      "AUTHORIZATIONFAILED" / "EXPIRED" /
                      "PRIVACYREQUIRED" / "CONTACTADMIN" / "NOPERM" /
                      "INUSE" / "EXPUNGEISSUED" / "CORRUPTION" /
                      "SERVERBUG" / "CLIENTBUG" / "CANNOT" / "LIMIT"
                      / "OVERQUOTA" / "ALREADYEXISTS" / "NONEXISTENT"
```


5. Security considerations

Revealing information about a passphrase to unauthenticated IMAP
clients has bad karma.

Response codes are easier to parse than human-readable text. This can
amplify the consequences of an information leak. For example,
selecting a mailbox can fail because the mailbox doesn't exist,
because the user doesn't have the "l" right (right to know the
mailbox exists) or "r" (right to read the mailbox). If the server
sent different responses in the first two cases in the past, only
malevolent clients would discover it. With response codes it's
possible, perhaps probable, that benevolent clients forward the
leaked information to the user. Server authors are encouraged to be
particularly careful with the NOPERM and authentication-related
responses.

6. **IANA considerations**

   The IANA is requested to create a new registry, tentatively named
   imap-response-codes, and populate it as follows:

        REFERRAL   RFC 2221
        ALERT   RFC 3501
        BADCHARSET   RFC 3501
        PARSE   RFC 3501
        PERMANENTFLAGS   RFC 3501
        READ-ONLY   RFC 3501
        READ-WRITE   RFC 3501
        TRYCREATE   RFC 3501
        UIDNEXT   RFC 3501
        UIDVALIDITY   RFC 3501
        UNSEEN   RFC 3501
        UNKNOWN-CTE   RFC 3516
        UIDNOTSTICKY   RFC 4315
        APPENDUID   RFC 4315
        COPYUID   RFC 4315
        URLMECH   RFC 4467
        TOOBIG   RFC 4469
        BADURL   RFC 4469
        HIGHESTMODSEQ   RFC 4551
        NOMODSEQ   RFC 4551
        MODIFIED   RFC 4551
        COMPRESSIONACTIVE   RFC 4978
        CLOSED   RFC 5162
        BADCOMPARATOR   RFC 5255
        ANNOTATE   RFC 5257
        METADATA   RFC (draft-daboo-imap-annotatemore-16.txt)
        UNAVAILABLE   RFC (this)
        AUTHENTICATIONFAILED RFC (this)
        AUTHORIZATIONFAILED RFC (this)
        EXPIRED   RFC (this)
        PRIVACYREQUIRED   RFC (this)
        CONTACTADMIN   RFC (this)
        NOPERM   RFC (this)
        INUSE   RFC (this)
        EXPUNGEISSUED   RFC (this)
        CORRUPTION   RFC (this)
        SERVERBUG   RFC (this)
        CLIENTBUG   RFC (this)
        CANNOT   RFC (this)
        LIMIT   RFC (this)
        OVERQUOTA   RFC (this)
        ALREADYEXISTS   RFC (this)
        NONEXISTENT   RFC (this)

The RFC editor is requested to delete this entire text, and insert a
sentence or two mentioning the registry's URL instead.

The new registry should only be extended by publishing an RFC. The
IANA may to add placeholders for internet-drafts at its discretion.

7. Acknowledgements

Peter Coates, Mark Crispin, Philip Guenther, Philip Van Hoof, Alexey
Melnikov, Ken Murchison, Chris Newman, Timo Sirainen, Dale Wiggins
and Sarah Wilkin helped with this document.

8. Normative References

[RFC3501]   Crispin, "Internet Message Access Protocol - Version
            4rev1", RFC 3501, University of Washington, June 2003.

[RFC5234]   Crocker, Overell, "Augmented BNF for Syntax
            Specifications: ABNF", RFC 5234, Brandenburg
            Internetworking, THUS plc, January 2008.

9. Informative References

[RFC2087]   Myers, "IMAP4 QUOTA extension", RFC 2087, Carnegie Mellon,
            January 1997.

[RFC2180]   Gahrns, "IMAP4 Multi-Accessed Mailbox Practice", RFC 2180,
            Microsoft, July 1997.

[RFC4314]   Melnikov, "IMAP4 Access Control List (ACL) Extension", RFC
            4314, December 2005.

10. Author's Address

Arnt Gulbrandsen
Oryx Mail Systems GmbH
Schweppermannstr. 8
D-81671 Muenchen
Germany

Fax: +49 89 4502 9758

Email: arnt@oryx.com

(RFC Editor: Please delete everything after this point)

Open Issues

   I took TOOWEAK out since it doesn't seem to have real purpose: "The
   server requires a stronger authentication mechanism. If the
   connection is not encrypted, the client could also try the same
   mechanism via an encrypted connection." But now I remember why it was
   there: The server may offer e.g. AUTH=CRAM-MD5, but not be able to
   carry that out for every user. Maybe it should be returned with a
   better name. I'd like to hear whether anyone actually does this.

   The name, if any, should reflect that the server cannot carry out
   this particular mechanism for this particular authentication-id. This
   may be because it's too weak (a policy decision) or because the
   server lacks data for this (user,mechanism) combination.

Changes since -00

   - CHILDMAILBOXEXISTS merged into INUSE.

   - ACCESSDENIED renamed ACL to clarify its scope.

   - NOBODYPART scheduled for deletion if noone minds.

   - EXISTS renamed ALREADYEXISTS to avoid confusion with the EXISTS
     response. Mustn't overload developer brains. (Do unto others.)

   - Added a security note about how response codes makes some
     information leaks worse.

   - A couple of open issues.

Changes since -01

   - Two people independently argued that merging ALREADYEXISTS and
     NONEXISTENT was bad because of RENAME. Open issue closed.

   - An example for each response code.

   - EXPUNGED renamed, see EXISTS above.

   - EXPUNGEISSUED semantics changed to be 2180-neutral. It should now
     be equally useful no matter which part of 2180 the server
     implements.

- CONTACTADMIN vs. ALERT, an open issue.

- Added an IANA considerations section registering every (?) response
  code defined so far.

- Added contact details to CONTACTADMIN, by request.

- Resolved the CA/SB/C issue: The three responses may be handled
  similarly by some clients, but they may equally well be handled
  differently, so they should not be folded.

Changes since -02

- Removed the contact details for CONTACTADMIN. I think that was
  creeping featuritis, not likely to be implemented.

- Removed NOBODYPART, noone suggested use for it.

- Edited CORRUPTION to suggest that detailed information belongs in
  the server logs. The client/user can bug the admin to look in the
  log, but expecting users to transmit information is stupid.

- Updated the IANA list for 5255 and 5257.

Changes since -03

- Explained the criteria for inclusion/exclusion better.

- Fixed remove/delete typo, fix status type

- Better text in the CANNOT example

- Instruct the IANA to extend the registry only when an RFC is
  published

Changes since -04

- ACL renamed NOPERM on request of Timo.

- Added METADATA, which I had overlooked.

- Turned ANNOTATE <VARIOUS> into just ANNOTATE, added METADATA
  <VARIOUS> in the same way.

Changes since -05

    - Fix typo (by rewriting the sentence)

Changes since -06

    - added PRIVACYREQUIRED

    - Random formatting, since this now close to RFC