

Network Working Group
Internet-Draft
Intended Status: Proposed Standard

Arnt Gulbrandsen
Oryx Mail Systems GmbH
Alexey Melnikov
Isode Limited
December 16, 2007

The IMAP ENABLE Extension
draft-gulbrandsen-imap-enable-05.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

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>.

This Internet-Draft expires in February 2008.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

Most IMAP extensions are used by the client when it wants to and the server supports it. However, a few extensions require the server to know whether a client supports that extension. The ENABLE extension allows an IMAP client to say which extensions it supports.

Internet-draft

December 2007

1. 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 [\[RFC2119\]](#).

Formal syntax is defined by [\[RFC4234\]](#) as modified by [\[RFC3501\]](#).

Example lines prefaced by "C:" are sent by the client and ones prefaced by "S:" by the server. The five characters [...] means that something has been elided.

2. Overview

Several IMAP extensions allow the server to return unsolicited responses specific to these extensions in certain circumstances. However, servers cannot send those unsolicited responses until they know that the clients support such extensions and thus won't choke on the extension response data.

Up until now extensions have typically stated that a server cannot send the unsolicited responses until after the client has used a command with the extension data (i.e. at that point the server knows the client is aware of the extension). CONSTORE ([\[RFC4551\]](#)), ANNOTATE ([\[ANNOTATE\]](#)) and some extensions under consideration at the moment use various commands to enable server extensions. For example CONDSTORE uses a SELECT or FETCH parameter, and ANNOTATE uses a side effect of FETCH.

The ENABLE extension provides an explicit indication from the client that it supports particular extensions. This is done using a new ENABLE command.

An IMAP server which supports ENABLE advertises this by including the word ENABLE in its capability list.

Most IMAP extensions do not require the client to enable the extension in any way.

3. Protocol changes

3.1 The ENABLE Command

Arguments: capability names

Result: OK: Relevant capabilities enabled
BAD: No arguments, or syntax error in an argument

Gulbrandsen

Expires June 2008

[Page 2]

Internet-draft

December 2007

The ENABLE command takes a list of capability names, and requests the server to enable the named extensions. Once enabled using ENABLE, each extension remains active until the IMAP connection is closed. For each argument, the server does the following:

- If the argument is not an extension known to the server, the server MUST ignore the argument.
- If the argument is an extension known to the server, and it is not specifically permitted to be enabled using ENABLE, the server MUST ignore the argument. (Note that knowing about an extension doesn't necessarily imply supporting that extension.)
- If the argument is an extension is supported by the server and which needs to be enabled, the server MUST enable the extension for the duration of the connection. At present this applies only to CONDSTORE ([[RFC4551](#)]). Note that once an extension is enabled, there is no way to disable it.

If the ENABLE command is successful, the server MUST send an untagged ENABLED response (see [Section 3.2](#)).

Clients SHOULD only include extensions that need to be enabled by the server. At the time this RFC is published CONDSTORE is the only such extension (ie. ENABLE CONDSTORE is an additional "Condstore enabling command" as defined in [[RFC4551](#)]). Future RFCs may add to this list. [Note to the RFC Editor: If the IMAP ANNOTATE document has been published already, ANNOTATE should be mentioned as well as CONDSTORE.]

The ENABLE command is only valid in Authenticated state (see [[RFC3501](#)]), before any mailbox is selected. Clients MUST NOT issue ENABLE once they SELECT/EXAMINE a mailbox, however server implementations don't have to check that no mailbox is selected or was previously selected during the duration of a connection.

The ENABLE command can be issued multiple times in a session. It is additive, i.e. "ENABLE a b", followed by "ENABLE c" is the same as a single command "ENABLE a b c". When multiple ENABLE commands are issued, each corresponding ENABLED response SHOULD only contain extensions enabled by the corresponding ENABLE command.

There are no limitations on pipelining ENABLE. For example, it is possible to send ENABLE and then immediately SELECT, or a LOGIN immediately followed by ENABLE.

The server MUST NOT change the CAPABILITY list as a result of executing ENABLE, i.e. a CAPABILITY command issued right after an

ENABLE command MUST list the same capabilities as a CAPABILITY command issued before the ENABLE command. The following example demonstrates that:

```
C: t1 CAPABILITY
S: * CAPABILITY IMAP4rev1 ID LITERAL+ ENABLE X-GOOD-IDEA
S: t1 OK foo
C: t2 ENABLE CONDSTORE X-GOOD-IDEA
S: * ENABLED X-GOOD-IDEA
S: t2 OK foo
C: t3 CAPABILITY
S: * CAPABILITY IMAP4rev1 ID LITERAL+ ENABLE X-GOOD-IDEA
S: t3 OK foo again
```

In the following example, the client enables CONDSTORE:

```
C: a1 ENABLE CONDSTORE
S: * ENABLED CONDSTORE
S: a1 OK Conditional Store enabled
```

[3.2](#) The ENABLED Response

Contents: capability listing

The ENABLED response occurs as a result of an ENABLE command. The capability listing contains a space-separated listing of capability names that the server supports and that were successfully enabled. The ENABLED response may contain no capabilities, which means that

no extensions listed by the client were successfully enabled.

[3.3](#) Note to designers of extensions that may use the ENABLE command

Designers of IMAP extensions are discouraged from creating extensions that require ENABLE unless there is no good alternative design. Specifically, extensions that cause potentially incompatible behavior changes to deployed server responses (and thus benefit from ENABLE) have a higher complexity cost than extensions that do not.

[4.](#) Formal Syntax

The following syntax specification uses the Augmented Backus-Naur Form (ABNF) notation as specified in [[RFC4234](#)] including the core rules in [Appendix B.1.](#) [[RFC3501](#)] defines the non-terminals "capability" and "command-any".

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.

```
capability    =/ "ENABLE"

command-any   =/ "ENABLE" 1*(SP capability)

response-data =/ "*" SP enable-data CRLF

enable-data   = "ENABLED" *(SP capability)
```

[5.](#) Security considerations

It is believed that this extension doesn't add any new security considerations that are not already present in the base IMAP protocol [[RFC3501](#)].

6. IANA considerations

The IANA is requested to add ENABLE to the IMAP4 Capabilities Registry. [[Note to RFC-editor: please remove the following before publication: This registration should take place at the following location: <http://www.iana.org/assignments/imap4-capabilities>]]

7. Acknowledgements

Editors would like to thank Randy Gellens, Chris Newman, Peter Coates, Dave Cridland, Mark Crispin, Ned Freed, Dan Karp, Cyrus Daboo, Ken Murchison and Eric Burger for comments and corrections. However this doesn't necessarily mean that they endorse this extension, agree with all details or responsible for errors introduced by editors.

8. Normative References

[RFC2119] Bradner, "Key words for use in RFCs to Indicate Requirement Levels", [RFC 2119](#), Harvard University, March 1997.

[RFC3501] Crispin, "Internet Message Access Protocol - Version

Gulbrandsen

Expires June 2008

[Page 5]

Internet-draft

December 2007

4rev1", [RFC 3501](#), University of Washington, June 2003.

[RFC4234] Crocker, Overell, "Augmented BNF for Syntax Specifications: ABNF", [RFC 4234](#), Brandenburg Internetworking, Demon Internet Ltd, October 2005.

[RFC4551] Melnikov, Hole, "IMAP Extension for Conditional STORE Operation or Quick Flag Changes Resynchronization", [RFC 4551](#), Isode Ltd., June 2006.

8. Informative References

[RFC2177] Leiba, "IMAP4 IDLE Command", [RFC 2177](#), IBM, June 1997.

[ANNOTATE] Daboo, Gellens, "IMAP ANNOTATE Extension", [draft-ietf-
imapext-annotate](#), August 2006.

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

Alexey Melnikov
Isode Ltd
5 Castle Business Village
36 Station Road
Hampton, Middlesex TW12 2BX
UK

Email: Alexey.Melnikov@isode.com

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information

on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Full Copyright Statement

Copyright (C) The IETF Trust (2007). This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.

Changes since -00

- The IANA asked me to specify the IANA registry exactly
- Say "clients should only use ENABLE when it's really necessary"
- Better abstract
- Wording.
- Refer to RFCs by number, not by topic.
- Boilerplate updates - IETF Trust and so on.

Changes since -01

- Specify that ENABLE ID is BAD, not ignorable.
- Explicitly allow maximum pipelining.
- Security implications.

Changes since -02

- Nits
- Unique tags in examples
- Note specifically that a server can reply BAD to ENABLE ID, even if it doesn't support ID. All it needs is to know that ID cannot be ENABLED.

Changes since -03

- Added ENABLED response as per discussion on the mailing list
- Changed ENABLE to never return BAD
- Only allow ENABLE in the authenticated state as per consensus in Vancouver
- Clarified [lack of] interaction with the CAPABILITY response

- Clarified that the ENABLE command is additive
- Added more examples

Changes since -04

- Added rationale for the ENABLE extension
- Fixed several inconsistencies caused by restringing ENABLE to authentication state only.

