IMAP4 ID extension

Status of this Memo

This document is an Internet-Draft. 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."

To learn the current status of any Internet-Draft, please check the
"id-abstracts.txt" listing contained in the Internet-Drafts
Shadow Directories on ftp.is.co.za (Africa), ftp.nordu.net (Europe),
munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or
ftp.isi.edu (US West Coast).

A revised version of this draft document will be submitted to the RFC
editor as a Proposed Standard for the Internet Community. Discussion
and suggestions for improvement are requested. This document will
expire before October 1997. Distribution of this draft is unlimited.

1. Abstract

The purpose of the ID extension to the IMAP4rev1 protocol is to allow
the server and client to exchange identification information on their
implementation in order to make bug reports and usage statistics more
complete.

Although the IMAP4rev1 protocol described in [IMAP4rev1] provides a
method for accessing remote mail stores, there is no facility to
advertise what program a client or server uses to provide service.
This makes it difficult for implementors to get complete bug reports
from users, as it is frequently difficult to know what client or
server is in use.

Additionally, some sites may wish to assemble usage statistics based
on what clients are used, but in an an environment where users are
permitted to obtain and maintain their own clients this is difficult
to accomplish.
The ID command provides a facility to advertise information on what programs are being used, along with contact information should bugs ever occur.

2. Conventions Used in this Document

The conventions used in this document are the same as specified in [IMAP4rev1]. In examples, "C:" and "S:" indicate lines sent by the client and server respectively. Line breaks have been inserted for readability.

3. Specification

The ID extension is provided in order for clients and servers to exchange information on their implementations.

This information may be submitted to a server by any client wishing to provide information for statistical purposes, provided the server advertises its willingness to take the information with the atom "ID" included in the list of capabilities returned by the CAPABILITY command.

Implementations MUST NOT make operational changes based on the data sent as part of the ID command or response.

3.1. ID Command

Arguments: client parameter list or NIL

Responses: OPTIONAL untagged response: ID

Result: OK identification information accepted
BAD command unknown or arguments invalid

Identification information is sent by the client with the ID command. The information sent is in the form of a list of field/value pairs. Fields are permitted to be any IMAP4 string, and values are permitted to be any IMAP4 string or NIL. A value of NIL indicates that the client can not or will not specify this information. The client may also send NIL instead of the list, indicating that it wants to send no information, but would still accept a server response.

The available fields are defined in section 3.3.

Example: C: a023 ID ("name" "sodr" "version" "19.34" "vendor" "Pink Floyd Music Limited")
S: a023 OK ID completed
3.2. ID Response

Contents: server parameter list

In response to an ID command issued by the client, the server MAY reply with a tagged response containing information on its implementation. The format is the same as the client list.

Example:

C: a023 ID NIL
S: * ID ("name" "Cyrus" "version" "1.5" "os" "sunos" "os-version" "5.5" "email" "cyrus-bugs@andrew.cmu.edu")
S: a023 OK ID command completed

The server is not required to send an ID response to an ID command.

3.3. Defined Field Values

Any string may be sent as a field, but the following are defined to describe certain values that might be sent. Implementations are free to send none, any, or all of these. Strings are not case-sensitive.

name                     Name of the program
version                  Version number of the program
os                       Name of the operating system
os-version               Version of the operating system
vendor                   Vendor of the client/server
contact                  Contact person for the client/server
email                    Email address of contact/vendor
phone                    Phone number of contact/vendor
address                  Postal address of contact/vendor
date                     Date program was released; should be in a human-readable form
command                  Command used to start the program
arguments                Arguments supplied on the command line, if any
environment              Description of environment, i.e., UNIX environment variables or Window registry settings.

Implementations MUST NOT use contact information to submit automatic bug reports. Information supplied with the ID command is not necessarily complete or accurate. Implementations may include such information in a report submitted.

It is preferrable to find the name and version of the underlying
Some of this information (specifically, the command, arguments, and environment fields) is likely to violate users’ privacy. See the section on security considerations below.

Implementations may send the same tag more than once:

Example: C: a001 ID ("client-name" NIL "client-name" "spooky" "client-name" NIL) S: a001 OK ID completed

4. Formal Syntax

This syntax is intended to augment the grammar specified in [IMAP4rev1] in order to provide for the ID command. This specification uses the augmented Backus-Naur Form (BNF) notation as specified in [RFC-822] and modified by [IMAP4rev1].

```plaintext
command_any ::= "CAPABILITY / "LOGOUT" / "NOOP" / x_command / id
               ;; adds id command to command_any in [IMAP4rev1]

id ::= "ID" SPACE id_params_list / nil

id_response ::= "ID" SPACE id_params_list / nil

id_params_list ::= "(#(string SPACE nstring) ")" / nil
                  ;; list of field value pairs

response_data ::= "*" (resp_cond_state / resp_cond_bye / mailbox_data / message_data / capability_data / id_response)
```

5. References


6. Security Considerations

A client or server supporting ID may violate the privacy of users by sending information on the underlying implementation, or in some cases, information on how the mail client was invoked. For this reason, it is highly desirable that client implementations provide a
method of disabling ID support -- by either not sending ID at all, or by sending the command "ID NIL", or by controlling the level of detail of information sent. A server may reply similarly in its untagged response (** ID NIL"), or it may send no untagged response.

7. Author’s Address

Tim Showalter
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

Email: tjs@andrew.cmu.edu