Sieve Email Filtering: Delivery by MAILBOXID

Abstract

The OBJECTID capability of IMAP (RFC 8474) allows clients to identify mailboxes by a unique identifier that survives renaming.

This document extends the Sieve email filtering language (RFC 5228) to allow using that same unique identifier as a target for fileinto rules and for testing the existence of mailboxes.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9042.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.
1. **Introduction**

Sieve rules [RFC5228] are sometimes created using graphical interfaces, which allow users to select the mailbox to be used as a target for a rule.

If that mailbox is renamed, the client may also update its internal representation of the rule and update the Sieve script to match; however, this is a multistep process and subject to partial failures. Also, if the folder is renamed by a different mechanism (e.g., another IMAP client), the rules will get out of sync.

By telling `fileinto` to reference the immutable `MAILBOXID` specified by [RFC8474], using the extension specified herein, Sieve rules can continue to target the same mailbox, even if it gets renamed.
2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Sieve Capability String

Scripts that use the extensions defined in this document MUST explicitly require the capability "mailboxid".

Example:

```plaintext
require "mailboxid";
```

4. Argument :mailboxid to Command fileinto

Normally, the fileinto command delivers the message in the mailbox specified using its positional mailbox argument. However, if the optional :mailboxid argument is also specified, the fileinto command first checks whether a mailbox exists in the user’s personal namespace [RFC2342] with the specified MAILBOXID [RFC8474].

If a matching mailbox is found, that mailbox is used for delivery.

If there is no such mailbox, the fileinto action proceeds as it would without the :mailboxid argument.

The tagged argument :mailboxid to fileinto consumes one additional token, a string containing the OBJECTID of the target mailbox.

Example:

```plaintext
require "fileinto";
require "mailboxid";

if header :contains ["from"] "coyote" {
    fileinto :mailboxid "F6352ae03-b7f5-463c-896f-d8b48ee3" "INBOX.harassment";
}
```
4.1. Interaction with Mailbox Extension

For servers that also support the mailbox extension defined in [RFC5490], if both the :create and :mailboxid arguments are provided to a fileinto command and no matching mailbox is found, then a new mailbox will be created.

This new mailbox will have the name specified by the positional mailbox argument ([RFC5228], Section 4.1); however, it will get a different MAILBOXID (chosen by the server) rather than the one specified by the :mailboxid argument to fileinto.

Example:

```
require "fileinto";
require "mailboxid";
require "mailbox";
fileinto :mailboxid "Fnosuch"
  :create
  "INBOX.no-such-folder";
  # creates INBOX.no-such-folder, but it doesn't
  # get the "Fnosuch" mailboxid.
```

4.2. Interaction with Special-Use Extension

For servers that also support delivery to special-use mailboxes [RFC8579], it is an error to specify both :mailboxid and :specialuse in the same fileinto command.

Advanced filtering based on both special-use and MAILBOXID can be built with explicit specialuse_exists and mailboxidexists tests.

```
Note to developers of Sieve generation tools:
It is advisable to use special-use rather than MAILBOXID when creating rules that
are based on a special-use purpose (e.g., delivery directly to the Junk folder based on
a header that was added by a scanning agent earlier in the mail flow).
```

5. Interaction with FCC Extension

This document extends the definition of the :fcc argument defined in [RFC8580] so that it can optionally be used with the :mailboxid argument. The syntax for FCC is extended here using ABNF [RFC5234]:

```
MAILBOXID-OPT = ":mailboxid" objectid
FCC-OPTS =/ MAILBOXID-OPT
```
If the optional :mailboxid argument is specified with :fcc, it instructs the Sieve interpreter to check whether a mailbox exists with the specific MAILBOXID. If such a mailbox exists, the generated message is filed into that mailbox. Otherwise, the generated message is filed into the :fcc target mailbox.

As with fileinto, it is an error to specify both :mailboxid and :specialuse for the same fcc rule.

Example:

```plaintext
require ["enotify", "fcc", "mailboxid"];  
notify :fcc "INBOX.Sent"  
    :mailboxid "F6352ae03-b7f5-463c-896f-d8b48ee3"  
    :message "You got mail!"  
    "mailto:ken@example.com";
```

6. Test mailboxidexists

Usage: mailboxidexists <mailbox-objects: string-list>

The mailboxidexists test is true if every string argument provided is the MAILBOXID of a mailbox that exists in the mailstore and that allows the user in whose context the Sieve script runs to deliver messages into it.

When the mailstore is an IMAP server that also supports IMAP Access Control List (ACL) [RFC4314], delivery is allowed if the user has the 'p' or 'i' rights for the mailbox (see Section 5.2 of [RFC4314]).

When the mailstore is an IMAP server that does not support IMAP ACL, delivery is allowed if the READ-WRITE response code is present for the mailbox when selected by the user (see Section 7.1 of [RFC3501]).

Note that a successful mailboxidexists test for a mailbox doesn't necessarily mean that a "fileinto :mailboxid" action on this mailbox would succeed. For example, the fileinto action might put the user over quota. The mailboxidexists test only verifies existence of the mailbox and whether the user in whose context the Sieve script runs has permissions to execute fileinto on it.
Example:

```plaintext
require "fileinto";
require "mailboxid";

if header :contains ["from"] "coyote" {
  if mailboxidexists "F6352ae03-b7f5-463c-896f-d8b48ee3" {
    fileinto :mailboxid "F6352ae03-b7f5-463c-896f-d8b48ee3"
      "INBOX.name.will.not.be.used";
  } else {
    fileinto "INBOX.harassment";
  }
}
```

Note to implementers:

This test behaves identically to the mailboxexists test defined in [RFC5490] but operates on MAILBOXIDs rather than mailbox names.

7. Interaction with Variables Extension

There is no special interaction defined; however, as an OBJECTID is a string in this document, OBJECTID values can contain variable expansions if [RFC5229] is enabled.

8. Security Considerations

Because MAILBOXID is always generated by the server, implementations MUST NOT allow Sieve to make an end run around this protection by creating mailboxes with the specified ID by using :create and :mailboxid in a fileinto rule for a nonexistent mailbox.

Implementers are referred to the Security Considerations sections of [RFC5228] and [RFC8474].

9. IANA Considerations

IANA has added the following capability to the "Sieve Extensions" registry at <https://www.iana.org/assignments/sieve-extensions>:

Capability name:  mailboxid
Description:  adds a test for checking mailbox existence by OBJECTID and new optional arguments to fileinto and :fcc that allow selecting the destination mailbox by OBJECTID.
RFC number:  RFC 9042
Contact address:  EXTRA discussion list <extra@ietf.org>
10. References

10.1. Normative References


10.2. Informative References


Acknowledgements

This document borrows heavily from [RFC5490] for the matching mailboxexists test and from [RFC8579] for an example of modifying the fileinto command.

Thanks to Ned Freed, Ken Murchison, and Alexey Melnikov for feedback on the EXTRA mailing list.

Author's Address

Bron Gondwana (EDITOR)
Fastmail
Level 2
114 William St
Melbourne VIC 3000
Australia
Email: brong@fastmailteam.com
URI: https://www.fastmail.com