
Stream: Internet Engineering Task Force (IETF)
RFC: [9671](#)
Category: Standards Track
Published: October 2024
ISSN: 2070-1721
Authors: K. Murchison R. Signes M. Horsfall
Fastmail Fastmail Fastmail

RFC 9671

Sieve Email Filtering: Extension for Processing Calendar Attachments

Abstract

This document describes the "processcalendar" extension to the Sieve email filtering language. The "processcalendar" extension gives Sieve the ability to process machine-readable calendar data that is encapsulated in an email message using Multipurpose Internet Mail Extensions (MIME).

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/rfc9671>.

Copyright Notice

Copyright (c) 2024 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
2. Conventions Used in This Document	3
3. Capability Identifier	3
4. Process Calendar Action	4
4.1. Allow Public Argument	4
4.2. Addresses Argument	5
4.3. Updates Only Argument	5
4.4. Calendar ID Argument	5
4.5. Delete Cancelled Argument	6
4.6. Organizers Argument	6
4.7. Outcome Argument	6
4.8. Reason Argument	6
4.9. Interaction with Other Sieve Actions	6
4.10. Examples	7
5. Security Considerations	7
6. Privacy Considerations	8
7. IANA Considerations	9
7.1. Registration of Sieve Extension	9
7.2. Registration of Sieve Action	9
8. References	9
8.1. Normative References	9
8.2. Informative References	10
Acknowledgments	11
Authors' Addresses	11

1. Introduction

Users frequently receive invites, replies, and cancellations for events, tasks, etc. via Internet mail messages. It is sometimes desirable to have such messages automatically parsed and the enclosed calendar data added to, updated on, or deleted from the user's calendars.

Typically, such messages are based on the iCalendar Message-Based Interoperability Protocol (iMIP) [RFC6047]. However, sometimes the enclosed iCalendar [RFC5545] data does not include an iCalendar Transport-Independent Interoperability Protocol (iTIP) method property (see [RFC5546], Section 1.4), or the enclosed data may be in some other machine-readable format (e.g., JSCalendar [RFC8984]).

This document defines an extension to the Sieve language [RFC5228] that enables scripts to process machine-readable calendar data that is encapsulated in an email message using MIME [RFC2045]. Specifically, this extension provides the ability to alter items on a user's calendars that are referenced in the encapsulated calendar data.

2. Conventions Used in This Document

Conventions for notations are as in Section 1.1 of [RFC5228], including use of the "Usage:" label for the definition of action and tagged arguments syntax.

This document uses terminology and concepts from iCalendar [RFC5545] and iTIP [RFC5546] to describe the processing of calendar data, but this extension can be used with any machine-readable calendar data format that can express similar concepts.

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. Capability Identifier

Sieve interpreters that implement this extension **MUST** have an identifier of "processcalendar" for use with the capability mechanism.

4. Process Calendar Action

```
Usage: processcalendar [ :allowpublic ]
                        [ :addresses <string-list> ]
                        [ :updatesonly / :calendarid <string> ]
                        [ :deletecancelled ]
                        [ :organizers <ext-list-name: string> ]
                        [ :outcome <variablename: string> ]
                        [ :reason <variablename: string> ]
```

The "processcalendar" action is used to parse encapsulated calendar data and perform the appropriate action based on the content. If the calendar data is malformed in any way, it **MUST** be ignored and no action is taken. Otherwise, calendar objects may be created, updated, or deleted from a given calendar.

This action can be used with or without the "extlists" extension [RFC6134]. When the "extlists" extension is enabled in a script using <require "extlists">, the script can use the :organizers argument (Section 4.6) in the "processcalendar" action as described below. When the "extlists" extension is not enabled, the :organizers argument **MUST NOT** be used and **MUST** cause an error according to [RFC5228].

This action can be used with or without the "variables" extension [RFC5229]. When the "variables" extension is enabled in a script using <require "variables">, the script can use the :outcome (Section 4.7) and :reason (Section 4.8) arguments in the "processcalendar" action as described below. When the "variables" extension is not enabled, the :outcome and :reason arguments **MUST NOT** be used and **MUST** cause an error according to [RFC5228].

If a mail message contains calendar data in multiple MIME [RFC2045] parts, this action **MUST** verify that the calendar data in each part are semantically equivalent to one another. If the data is found to be semantically different, the action **MUST NOT** process the message. Otherwise, the action **MUST** only process one representation of the data.

This action **MUST NOT** make any changes to the participant status of the recipient when processing the calendar data. The mechanism for a recipient to change their participant status to an event is out of scope for this document.

This action **SHOULD** remove alarms from calendar data before applying it to a calendar. Failure to do so could result in unwelcome notifications being triggered for the recipient.

4.1. Allow Public Argument

The optional :allowpublic argument is used to tell the implementation that it can process calendar data that does not contain any ATTENDEE properties, such as iTIP messages where the METHOD is PUBLISH or non-iTIP messages where the calendar data does not contain METHOD and/or ORGANIZER properties.

If used in conjunction with the `:organizers` argument ([Section 4.6](#)), the implementation **MUST NOT** process non-iTIP messages.

If `:allowpublic` is omitted, the implementation **MUST NOT** process calendar data unless it is a well-formed iTIP message and one of the recipient user's email addresses matches the Calendar User Address (see [Section 3.3.3](#) of [\[RFC5545\]](#)) of the intended target of the message, as determined by the iTIP method (see [Section 1.4](#) of [\[RFC5546\]](#)) of the message:

- "REPLY": Value of the ORGANIZER property (see [Section 3.8.4.3](#) of [\[RFC5545\]](#))
- "REQUEST", "CANCEL", "ADD": Value of one of the ATTENDEE properties (see [Section 3.8.4.1](#) of [\[RFC5545\]](#))

The recipient user's email address matches the Calendar User Address of the target if the Calendar User Address is in the form of a mailto URI and the email address matches the "addr-spec" of the URI.

An email address is considered to belong to the recipient if it is one of the following:

- an email address known by the implementation to be associated with the recipient,
- the final envelope recipient address if it's available to the implementation, or
- an address specified by the script writer via the `:addresses` argument ([Section 4.2](#)).

4.2. Addresses Argument

The optional `:addresses` argument is used to specify email addresses that belong to the recipient in addition to the addresses known to the implementation.

4.3. Updates Only Argument

The optional `:updatesonly` argument is used to limit the messages processed to those targeting existing calendar objects only. If the message contains a new calendar object (its unique identifier does not exist on any of the user's calendars), the implementation **MUST NOT** add the object to a calendar.

If `:updatesonly` is omitted, new calendar objects may be added to one of the user's calendars.

The `:updatesonly` and `:calendarid` ([Section 4.4](#)) arguments are incompatible with each other. It is an error if both arguments are used in the same "processcalendar" action.

4.4. Calendar ID Argument

The optional `:calendarid` argument specifies the identifier of the calendar onto which new calendar objects should be placed.

If `:calendarid` is omitted, new calendar objects will be placed on the user's "default" calendar as determined by the implementation.

The `:updatesonly` (Section 4.3) and `:calendarid` arguments are incompatible with each other. It is an error if both arguments are used in the same "processcalendar" action.

4.5. Delete Cancelled Argument

The optional `:deletecancelled` argument is used to tell the implementation that if it receives a cancellation message, it **SHOULD** remove the associated calendar object from the calendar.

If `:deletecancelled` is omitted, the status of the associated calendar object will be set to cancelled and will remain on the calendar.

4.6. Organizers Argument

The optional `:organizers` argument is used to specify an external list of email addresses from which the recipient is willing to accept public events, invites, updates, and cancellations. Implementations **MUST NOT** process calendar data unless it is a well-formed iTIP message and one of the addresses in the external list matches the Calendar User Address of the ORGANIZER property. An email address in the external list matches the Calendar User Address of the ORGANIZER property if it is in the form of a mailto URI and the email address matches the "addr-spec" of the URI.

If `:organizers` is omitted, no validation of the ORGANIZER property is performed.

4.7. Outcome Argument

The optional `:outcome` argument specifies the name of a variable into which one of the following strings specifying the outcome of the action will be stored:

"no_action": No action was performed (e.g., the message didn't contain calendar data, or the set of provided options prevented the message from being processed).

"added": A new calendar object was added to a calendar.

"updated": A calendar object was updated, cancelled, or removed from the calendar.

"error": The message would have been processed but encountered an error in doing so.

4.8. Reason Argument

The optional `:reason` argument specifies the name of a variable into which a string describing the reason for the outcome will be stored. If no reason for the outcome is available, implementations **MUST** set the variable to the empty string.

For example, an outcome of "no_action" may have a reason of "only processing updates", or an outcome of "error" may have a reason of "missing unique identifier".

4.9. Interaction with Other Sieve Actions

The "processcalendar" action does not cancel Sieve's implicit keep action.

The "processcalendar" action can only be executed once per script. A script **MUST** fail with an appropriate error if it attempts to execute two or more "processcalendar" actions.

The "processcalendar" action is incompatible with the Sieve "reject" and "ereject" actions [RFC5429].

4.10. Examples

The following example specifies email addresses belonging to the user and the identifier of the calendar onto which to place new calendar objects:

```
require [ "processcalendar" ];

processcalendar :addresses [ "me@example.com", "alsome@example.com" ]
                  :calendarid "1ea6d86b-6c7f-48a2-bed3-2a4c40ec281a";
```

The following example tells the interpreter to process flight itineraries from a particular airline:

```
require [ "processcalendar" ];

if allof (address ["from", "sender"] "airline@example.com",
           header :contains "subject" "itinerary") {
    processcalendar :allowpublic;
}
```

The following example adds headers to the message if calendar data isn't processed :

```
require [ "processcalendar", "variables", "editheader" ];

set "processcal_outcome" "no_action";
set "processcal_reason" "";

processcalendar :outcome "processcal_outcome"
                  :reason "processcal_reason";

if not string :is "${processcal_outcome}" ["added", "updated"] {
    addheader "X-ProcessCal-Outcome" "${processcal_outcome}";
    addheader "X-ProcessCal-Reason" "${processcal_reason}";
}
```

5. Security Considerations

This document describes a method for altering an electronic calendar without user interaction. As such, unless proper precautions are undertaken, it can be used as a vector for calendar abuse.

It is critical that implementations correctly implement the behavior and restrictions described throughout this document. Security issues associated with processing unsolicited calendar data and methods for mitigating them are discussed in [CALSPAM]. Specifically:

- The "processcalendar" extension **MUST NOT** process any calendar data enclosed in a message flagged as spam and/or malicious. The "spamtest" and "virustest" extensions [RFC5235] (or the header test [RFC5228] if messages are scanned outside of the Sieve interpreter) can be used to make "processcalendar" conditional on "safe" content.
- The "processcalendar" extension **SHOULD NOT** process calendar data received from a potentially malicious sender. The address and envelope tests [RFC5228] (optionally along with the "extlists" extension [RFC6134]) can be used to create a "deny list" and make "processcalendar" conditional on the sender not being a member of that list.
- Similarly, the "processcalendar" extension **SHOULD** only process calendar data received from a known sender. The address and envelope tests [RFC5228] (optionally along with the "extlists" extension [RFC6134]) can be used to create an "allow list" and make "processcalendar" conditional on the sender being a member of that list.
- The "processcalendar" extension **SHOULD NOT** process calendar data received from an untrustworthy sender. Trustworthiness may depend on whether the message has a valid signature (see [RFC8551]) and/or on whether one or more of the following passes or fails on the message: Sender Policy Framework (SPF) [RFC7208], DomainKeys Identified Mail (DKIM) Signatures [RFC6376], and Domain-based Message Authentication, Reporting, and Conformance (DMARC) [RFC7489]. The mechanism by which a Sieve interpreter accesses the results of such checks is outside the scope of this document, but if the results are available in the message's header fields, the header test [RFC5228] can be used to make "processcalendar" conditional on the sender being trustworthy.

Additionally, if the calendar data has embedded (a.k.a. inline) attachments, implementations **SHOULD**:

- Decode the embedded attachment, if necessary.
- Scan the (decoded) attachment for malicious content.

If an attachment is found to be malicious, "processcalendar" **MUST NOT** process the calendar data.

6. Privacy Considerations

It is believed that this extension doesn't introduce any privacy considerations beyond those in [RFC5228].

7. IANA Considerations

7.1. Registration of Sieve Extension

This document defines the following new Sieve extension, which IANA has added to the "[Sieve Extensions](#)" registry. The registry is defined in [Section 6.2](#) of [\[RFC5228\]](#).

Capability name: processcalendar

Description: Adds the "processcalendar" action command to add and update items on a user's calendars.

RFC number: RFC 9671

Contact address: Sieve discussion list <sieve@ietf.org>

7.2. Registration of Sieve Action

This document defines the following new Sieve action, which IANA has added to the "[Sieve Actions](#)" registry. The registry is defined in [Section 2.1](#) of [\[RFC9122\]](#).

Name: processcalendar

Description: Add and update items on a user's calendars

References: RFC 9671 [\[RFC5229\]](#) [\[RFC6134\]](#)

Capabilities: "processcalendar", "variables", "extlists"

Action Interactions: This action is incompatible with the "reject" and "ereject" actions.

Cancels Implicit Keep? No

Can Use with IMAP Events? No

8. References

8.1. Normative References

[CALSPAM] The Calendaring and Scheduling Consortium, "Calendar operator practices - Guidelines to protect against calendar abuse", CC/R 18003:2019, 2019, <<https://standards.calconnect.org/csd/cc-18003.html>>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

- [RFC5228] Guenther, P., Ed. and T. Showalter, Ed., "Sieve: An Email Filtering Language", RFC 5228, DOI 10.17487/RFC5228, January 2008, <<https://www.rfc-editor.org/info/rfc5228>>.
- [RFC5229] Homme, K., "Sieve Email Filtering: Variables Extension", RFC 5229, DOI 10.17487/RFC5229, January 2008, <<https://www.rfc-editor.org/info/rfc5229>>.
- [RFC6047] Melnikov, A., Ed., "iCalendar Message-Based Interoperability Protocol (iMIP)", RFC 6047, DOI 10.17487/RFC6047, December 2010, <<https://www.rfc-editor.org/info/rfc6047>>.
- [RFC6134] Melnikov, A. and B. Leiba, "Sieve Extension: Externally Stored Lists", RFC 6134, DOI 10.17487/RFC6134, July 2011, <<https://www.rfc-editor.org/info/rfc6134>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC9122] Melnikov, A. and K. Murchison, "IANA Registry for Sieve Actions", RFC 9122, DOI 10.17487/RFC9122, June 2023, <<https://www.rfc-editor.org/info/rfc9122>>.

8.2. Informative References

- [RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, DOI 10.17487/RFC2045, November 1996, <<https://www.rfc-editor.org/info/rfc2045>>.
- [RFC5235] Daboo, C., "Sieve Email Filtering: Spamtest and Virustest Extensions", RFC 5235, DOI 10.17487/RFC5235, January 2008, <<https://www.rfc-editor.org/info/rfc5235>>.
- [RFC5429] Stone, A., Ed., "Sieve Email Filtering: Reject and Extended Reject Extensions", RFC 5429, DOI 10.17487/RFC5429, March 2009, <<https://www.rfc-editor.org/info/rfc5429>>.
- [RFC5545] Desruisseaux, B., Ed., "Internet Calendaring and Scheduling Core Object Specification (iCalendar)", RFC 5545, DOI 10.17487/RFC5545, September 2009, <<https://www.rfc-editor.org/info/rfc5545>>.
- [RFC5546] Daboo, C., Ed., "iCalendar Transport-Independent Interoperability Protocol (iTIP)", RFC 5546, DOI 10.17487/RFC5546, December 2009, <<https://www.rfc-editor.org/info/rfc5546>>.
- [RFC6376] Crocker, D., Ed., Hansen, T., Ed., and M. Kucherawy, Ed., "DomainKeys Identified Mail (DKIM) Signatures", STD 76, RFC 6376, DOI 10.17487/RFC6376, September 2011, <<https://www.rfc-editor.org/info/rfc6376>>.
- [RFC7208] Kitterman, S., "Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1", RFC 7208, DOI 10.17487/RFC7208, April 2014, <<https://www.rfc-editor.org/info/rfc7208>>.

- [RFC7489] Kucherawy, M., Ed. and E. Zwicky, Ed., "Domain-based Message Authentication, Reporting, and Conformance (DMARC)", RFC 7489, DOI 10.17487/RFC7489, March 2015, <<https://www.rfc-editor.org/info/rfc7489>>.
- [RFC8551] Schaad, J., Ramsdell, B., and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification", RFC 8551, DOI 10.17487/RFC8551, April 2019, <<https://www.rfc-editor.org/info/rfc8551>>.
- [RFC8984] Jenkins, N. and R. Stepanek, "JSCalendar: A JSON Representation of Calendar Data", RFC 8984, DOI 10.17487/RFC8984, July 2021, <<https://www.rfc-editor.org/info/rfc8984>>.

Acknowledgments

The authors would like to thank the following individuals for contributing their ideas and support for writing this specification: Ned Freed and Alexey Melnikov.

Authors' Addresses

Kenneth Murchison

Fastmail US LLC
1429 Walnut Street, Suite 1201
Philadelphia, PA 19102
United States of America
Email: murch@fastmailteam.com

Ricardo Signes

Fastmail US LLC
1429 Walnut Street, Suite 1201
Philadelphia, PA 19102
United States of America
Email: rjbs@fastmailteam.com

Matthew Horsfall

Fastmail US LLC
1429 Walnut Street, Suite 1201
Philadelphia, PA 19102
United States of America
Email: alh@fastmailteam.com