

**SIEVE Email Filtering: Include Extension**  
**draft-daboo-sieve-include-05**

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/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on December 27, 2006.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

The SIEVE Email Filtering "include" extension permits users to include one SIEVE script inside another. This can make managing large scripts or multiple sets of scripts much easier, as well as supporting common 'libraries' of scripts. Users are able to include their own personal scripts or site-wide scripts provided by the local SIEVE implementation.

Change History (to be removed prior to publication as an RFC)

Changes from -04 to -05:

- a. Fixed examples.
- b. Relaxed requirement that imported/exported variables be set before being used.

Changes from -03 to -04:

- a. Fixed missing 2119 definitions.
- b. Defined interaction with variables through use of import and export commands.

Changes from -02 to -03:

- a. Refreshing expired draft (updated for nits).
- b. Syntax -> Usage.
- c. Updated to 3028bis reference.

Changes from -01 to -02:

- a. Minor formatting changes only - refreshing expired draft.

Changes from -00 to -01:

- a. Added IPR boiler plate.
- b. Re-ordered sections at start to conform to RFC style.
- c. Moved recursion comment into General Considerations section.
- d. Switched to using optional parameter to indicate personal vs global.
- e. Explicitly state that an error occurs when a missing script is included.

Open Issues (to be resolved prior to publication as an RFC)

- a. Interaction with variables (scoping). Should variables be carried over between scripts that are included? Or should variables defined in an included script be local to that script only?



## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction and Overview . . . . .</a>	<a href="#">4</a>
<a href="#">2.</a>	<a href="#">Conventions Used in This Document . . . . .</a>	<a href="#">4</a>
<a href="#">3.</a>	<a href="#">SIEVE Include Extension . . . . .</a>	<a href="#">4</a>
<a href="#">3.1.</a>	<a href="#">General Considerations . . . . .</a>	<a href="#">4</a>
<a href="#">3.2.</a>	<a href="#">Control Structure Include . . . . .</a>	<a href="#">5</a>
<a href="#">3.3.</a>	<a href="#">Control Structure Return . . . . .</a>	<a href="#">8</a>
<a href="#">3.4.</a>	<a href="#">Interaction with Variables . . . . .</a>	<a href="#">8</a>
<a href="#">3.4.1.</a>	<a href="#">Control Structure Import . . . . .</a>	<a href="#">9</a>
<a href="#">3.4.2.</a>	<a href="#">Control Structure Export . . . . .</a>	<a href="#">9</a>
<a href="#">4.</a>	<a href="#">Security Considerations . . . . .</a>	<a href="#">10</a>
<a href="#">5.</a>	<a href="#">IANA Considerations . . . . .</a>	<a href="#">10</a>
<a href="#">5.1.</a>	<a href="#">include registration . . . . .</a>	<a href="#">11</a>
<a href="#">6.</a>	<a href="#">Normative References . . . . .</a>	<a href="#">11</a>
<a href="#">Appendix A.</a>	<a href="#">Acknowledgments . . . . .</a>	<a href="#">11</a>
	<a href="#">Author's Address . . . . .</a>	<a href="#">12</a>
	<a href="#">Intellectual Property and Copyright Statements . . . . .</a>	<a href="#">13</a>



## **1. Introduction and Overview**

It's convenient to be able to break SIEVE [[I-D.ietf-sieve-3028bis](#)] scripts down into smaller components which can be reused in a variety of different circumstances. For example, users may want to have a default script and a special 'vacation' script, the later being activated when the user goes on vacation. In that case the default actions should continue to be run, but a vacation command should be executed first. One option is to edit the default script to add or remove the vacation command as needed. Another is to have a vacation script that simply has a vacation command and then includes the default script.

## **2. Conventions Used in This Document**

Conventions for notations are as in SIEVE [[I-D.ietf-sieve-3028bis](#)] [section 1.1](#).

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](#)].

## **3. SIEVE Include Extension**

### **3.1. General Considerations**

SIEVE implementations that implement the "include", "return", "export" and "import" commands described below have an identifier of "include" for use with the capability mechanism. If any of the "include", "return", "export" or "import" commands are used in a script, the "include" capability MUST be listed in the "require" statement in that script.

SIEVE implementations must track the use of actions in included scripts so that implicit "keep" behavior can be properly determined based on whether any actions have executed in any script.

SIEVE implementations are allowed to limit the total number of nested included scripts, but MUST provide for a total of at least three levels of nested scripts including the top-level script. An error MUST be generated either when the script is uploaded to the SIEVE repository, or when the script is executed, if any nesting limit is exceeded. If such an error is detected whilst processing a SIEVE script, an implicit "keep" action MUST be executed to prevent loss of any messages.

Daboo

Expires December 27, 2006

[Page 4]

SIEVE implementations MUST ensure that recursive includes are not possible. i.e. if script "A" includes script "B", and script "B" includes script "A" an error MUST be generated either when the script is uploaded to the SIEVE repository, or when the script is executed. If such an error is detected whilst processing a SIEVE script, an implicit "keep" action MUST be executed to prevent loss of any messages.

SIEVE implementations MUST handle missing scripts being referenced via an includes in an existing script. An error MUST be generated when a missing included script is discovered during execution. If such an error is detected an implicit "keep" action MUST be executed to prevent loss of any messages.

If the SIEVE "variables" extension [[I-D.ietf-sieve-variables](#)] is present, an issue arises with the "scope" of variables defined in scripts that may include each other. For example, if a script defines the variable "\${status}" with one particular meaning or usage, and another defines "\${status}" with a different meaning, then if one script includes the other there is an issue as to which "\${status}" is being referenced. To solve this problem, SIEVE implementations MUST follow the scoping rules defined in [Section 3.4](#) and support the "import" and "export" commands defined there.

### **[3.2.](#) Control Structure Include**

Usage: include [[LOCATION](#)] <value: string>

The "include" command includes an optional parameter, and a single string argument representing the name of the script to include in the main script at that point.

[LOCATION] is an optional parameter that has one of two values:

Usage: ":personal" / ":global"

If the [[LOCATION](#)] parameter is not present, the location defaults to :personal.

The location has the following meanings:

:personal

Indicates that the named script is stored in the user's own personal (private) SIEVE repository.





:global

Indicates that the named script is stored in a site-wide SIEVE repository, accessible to all users of the SIEVE system.

The included script **MUST** be a valid SIEVE script, including having necessary "require" statements for all optional capabilities used by the script. The scope of a "require" statement in an included script is for that script only, not the including script. e.g. if script "A" includes script "B", and script "B" uses the "fileinto" extension, script "B" must have a "require" statement for "fileinto", irrespective of whether script "A" has one. In addition, if script "A" does not have a "require" statement for "fileinto", "fileinto" cannot be used anywhere in script "A", even after inclusion of script "B".

A "stop" command in an included script **MUST** stop all script processing, including the processing of the scripts that include the current one. The "return" command (described below) stops processing of the current script only, and allows the scripts that include it to continue.

Examples:

In the examples below, script content is indicated by a '|' as the first non-space character on a line for clarity. The '|' characters are not part of the script itself.

The user has four scripts stored in their personal repository:

"default"

This is the default active script that includes several others.

```
| require ["include"];  
|  
| include :personal "always_allow";  
| include :global "spam_tests";  
| include :personal "my_spam_tests";  
| include :personal "mailing_lists";
```

"always\_allow"

This script special cases some correspondent email addresses and makes sure any message containing those addresses are always kept.

Daboo

Expires December 27, 2006

[Page 6]

```
|   if header :is "From" "boss@example.com"
|   {
|       keep;
|   }
|   elsif header :is "From" "ceo@example.com"
|   {
|       keep;
|   }
```

"my\_spam\_tests"

This script does some user-specific spam tests to catch spam messages not caught by the site-wide spam tests.

```
|   require ["reject"];
|
|   if header :contains "Subject" "XXXX"
|   {
|       reject;
|   }
|   elsif header :is "From" "money@example.com"
|   {
|       reject;
|   }
```

"mailing\_lists"

This script looks for messages from different mailing lists and files each into a mailbox specific to the mailing list.

```
|   require ["fileinto"];
|
|   if header :is "Sender" "owner-ietf-mta-filters@imc.org"
|   {
|       fileinto "lists.sieve";
|   }
|   elsif header :is "Sender" "owner-ietf-imapext@imc.org"
|   {
|       fileinto "lists.imapext";
|   }
```

There is one script stored in the global repository:

"spam\_tests"

This script does some site-wide spam tests which any user at the site can include in their own scripts at a suitable point. The script content is kept up to date by the site administrator.



```
|   require ["reject"];
|
|   if anyof (header :contains "Subject" "$$",
|             header :contains "Subject" "Make money")
|   {
|       reject;
|   }
```

The "include" command may appear anywhere in the script where a control structure is legal.

Example:

```
|   require ["include"];
|
|   if anyof (header :contains "Subject" "$$",
|             header :contains "Subject" "Make money")
|   {
|       include "my_reject_script";
|   }
```

### [3.3.](#) Control Structure Return

Usage: return

The "return" command stops processing of the currently included script only and returns processing control to the script which includes it. If used in the main script (i.e. not in an included script), it has the same effect as the "stop" command, including the appropriate "keep" action if no other actions have been executed up to that point.

### [3.4.](#) Interaction with Variables

In order to avoid problems of variables in an included script "overwriting" those from the script that includes it, this specification requires that all variables defined in a script MUST be kept "private" to that script by default - i.e. they are not "visible" to other scripts. This ensures that two script authors cannot inadvertently cause problems by choosing the same name for a variable.

However, sometimes there is a need to make a variable defined in one script available to others. This specification defines the new commands "export" and "import" to alter the default behavior of variable scoping to allow variables to be "seen" by other scripts. The "export" command takes a list of variable names defined in a script and makes those available to any script that explicitly wants



to use them. The "import" command allows a script to gain access to variables that have been explicitly made available by other scripts. The explicit use of "export" and "import", coupled with the default behavior of variables only having local scope, ensures that multiple scripts cannot inadvertently overwrite each others variables.

#### **3.4.1. Control Structure Import**

Usage:    import <value: string-list>

The "import" command contains a string list argument that defines one or more variables names of variables exported by other scripts which should be made available to the current script.

The "import" command, if present, MUST be used immediately after any "require" commands (at least one of which will be present listing the "include" extension). Multiple "import" commands are allowed. An error occurs if an "import" command appears after a command other than "require" or "import". Use of the "import" command makes the listed variables immediately available for use in the body of the script that uses it.

If an "import" command lists a variable that has not been exported by any other script at that point during the SIEVE execution process, then an error MUST occur.

Example:

```
|   require ["variables", "fileinto", "include"];
|   import "test";
|
|   if header :contains "Subject" "${test}"
|   {
|       fileinto "INBOX.spam-${test};
|   }
```

#### **3.4.2. Control Structure Export**

Usage:    export <value: string-list>

The "export" command contains a string list argument that defines one or more variables names of variables defined in the current script which should be made available to any scripts that run during the current SIEVE script execution process.

The "export" command, if present, MUST be used immediately after any "require" or "import" commands. Multiple "export" commands are allowed. An error occurs if an "export" command appears after a





command other than "require", "import" or "export".

An error occurs if an "export" command lists a variable that is listed in an "import" command preceding it in the current script.

Example:

```
| require ["variables", "include"];
| export "test";
|
| set "test" "$$"
| include "spam_filter_script";
|
| set "test" "Make money"
| include "spam_filter_script";
```

Example:

```
| require ["variables", "include"];
| import "test";
| export "test-mailbox";
|
| if header :contains "Subject" "${test}"
| {
|     set "test-mailbox" "INBOX.spam-${test}";
|     include "fileinto-script"
| }
```

#### **4. Security Considerations**

SIEVE implementations MUST ensure adequate security for the global script repository to prevent unauthorized changes to global scripts.

Beyond that, the "include" extension does not raise any security considerations that are not present in the base SIEVE protocol, and these issues are discussed in SIEVE.

#### **5. IANA Considerations**

The following template specifies the IANA registration of the Sieve extension specified in this document:



### **5.1. include registration**

To: iana@iana.org  
Subject: Registration of new Sieve extension

Capability name: include  
Capability keyword: include  
Capability arguments: N/A  
Standards Track/IESG-approved experimental RFC number: this RFC  
Person and email address to contact for further information:

Cyrus Daboo

<mailto:cyrus@daboo.name>

This information should be added to the list of sieve extensions  
given on <http://www.iana.org/assignments/sieve-extensions>.

This information should be added to the list of sieve extensions  
given on <http://www.iana.org/assignments/sieve-extensions>.

## **6. Normative References**

- [I-D.ietf-sieve-3028bis]  
Showalter, T. and P. Guenther, "Sieve: An Email Filtering  
Language", [draft-ietf-sieve-3028bis-06](#) (work in progress),  
March 2006.
- [I-D.ietf-sieve-variables]  
Homme, K., "Sieve Extension: Variables",  
[draft-ietf-sieve-variables-08](#) (work in progress),  
December 2005.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate  
Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

## **Appendix A. Acknowledgments**

Thanks to Aaron Stone, Ken Murchison, Rob Siemborski, Alexey  
Melnikov, Marc Mutz and Kjetil Torgrim Homme for comments and  
corrections.



Author's Address

Cyrus Daboo

Email: [cyrus@daboo.name](mailto:cyrus@daboo.name)

## 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](mailto:ietf-ipr@ietf.org).

## Disclaimer of Validity

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 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.

## Copyright Statement

Copyright (C) The Internet Society (2006). 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.

## Acknowledgment

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

