

SIEVE Include Extension

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

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>.

Copyright Notice

Copyright (C) The Internet Society 2003. All Rights Reserved.

Daboo

Expires October 2003

[Page 1]

Table of Contents

1	Abstract	2
2	Conventions Used in This Document	2
3	Change History	2
4	Open Issues:	2
5	Introduction and Overview	2
6	SIEVE Include Extension	3
6.1	General Considerations	3
6.2	Control Structure Include	3
6.3	Control Structure Return	6
7	Security Considerations	6
8	IANA Considerations	6
8.1	include registration	6
9	Normative References	7
10	Acknowledgments	7
11	Author's Address	7
12	Full Copyright Statement	7

[1](#) Abstract

The SIEVE [[SIEVE](#)] "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.

[2](#) Conventions Used in This Document

Conventions for notations are as in [[SIEVE](#)] [section 1.1](#), including use of [[KEYWORDS](#)].

[3](#) Change History

Initial Document

[4](#) Open Issues:

- 1 Should we allow URIs to point to scripts to include?
- 2 Should we allow string lists as argument to include?
- 3 How should non-exist include scripts be handled?
- 4 Interaction with variables?

[5](#) Introduction and Overview

Its convenient to be able to break SIEVE 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.

6 SIEVE Include Extension

6.1 General Considerations

SIEVE implementations that implement the "include" and "return" control structures described below have an identifier of "include" for use with the capability mechanism. If either of the "include" or "return" control structures 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" behaviour 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.

6.2 Control Structure Include

Syntax: include <value: string>

The "include" control structure includes a single string argument representing the name of the script to include in the main script at that point.

The script name contains one of two prefixes:

"/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 second component of the name represents an implementation dependent script name.

Daboo

Expires October 2003

[Page 3]

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

Examples:

In the example 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.

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

Daboo

Expires October 2003

[Page 4]

```
"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-imapect@imc.org"
|   {
|       fileinto "lists.imapect";
|   }
```

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

Daboo

Expires October 2003

[Page 5]

Example:

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

[6.3](#) Control Structure Return

Syntax: return

The "return" control structure 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" control structure, including the appropriate "keep" action if no other actions have been executed up to that point.

[7](#) Security Considerations

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 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](#)].

[8](#) IANA Considerations

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

[8.1](#) include registration

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

Capability name: include

Capability keyword: include
Capability arguments: N/A

Daboo

Expires October 2003

[Page 6]

Standards Track/IESG-approved experimental RFC number: this RFC
Person and email address to contact for further information:

Cyrus Daboo
Cyrusoft International, Inc.
Suite 780, 5001 Baum Blvd.
Pittsburgh, PA 15213
U.S.A.

Email: daboo@cyrusoft.com

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

9 Normative References

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

[SIEVE] Showalter, "Sieve: A Mail Filtering Language", [RFC 3028](#), January 2001.

10 Acknowledgments

Thanks to Ken Murchison and Rob Siemborski for comments and suggestions.

11 Author's Address

Cyrus Daboo
Cyrusoft International, Inc.
Suite 780, 5001 Baum Blvd.
Pittsburgh, PA 15213
U.S.A.

Email: daboo@cyrusoft.com

12 Full Copyright Statement

Copyright (C) The Internet Society 2003. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph

are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing

the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS 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.

Daboo

Expires October 2003

[Page 8]