

Groupe de travail Réseau
Request for Comments : 5229
 RFC mise à jour : 5228
 Catégorie : Sur la voie de la normalisation

K. Homme, University of Oslo
 janvier 2008

Traduction Claude Brière de L'Isle

Filtrage de messagerie Sieve : extension de variables

Statut du présent mémoire

Le présent document spécifie un protocole Internet sur la voie de la normalisation pour la communauté de l'Internet, et appelle à des discussions et suggestions pour son amélioration. Prière de se référer à l'édition en cours des "Normes officielles des protocoles de l'Internet" (STD 1) pour connaître l'état de la normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

Résumé

Dans les ensembles de règles de filtrage de messagerie évoluées, il est utile de conserver l'état ou les détails de configuration à travers les règles. Le présent document met à jour le langage de filtrage Sieve (RFC 5228) avec une extension pour prendre en charge les variables. L'extension change l'interprétation des chaînes, ajoute une action pour mémoriser les données dans les variables, et fournit un nouvel essai afin que la valeur d'une chaîne puisse être examinée.

Table des Matières

1. Introduction.....	1
2. Identifiant de capacité.....	1
3. Interprétation des chaînes.....	2
3.1 Caractères de citation et codés.....	2
3.2 Variables de correspondance.....	3
4. Action set.....	3
4.1 Modificateurs.....	4
5. Chaîne d'essai.....	5
6. Limites de mise en œuvre.....	5
7. Considérations sur la sécurité.....	6
8. Considérations relatives à l'IANA.....	6
9. Remerciements.....	6
10. Références.....	6
10.1 Références normatives.....	6
10.2 Références pour information.....	7
Adresse de l'auteur.....	7
Déclaration complète de droits de reproduction.....	7

1. Introduction

Ceci est une extension au langage Sieve défini par la [RFC5228]. Elle ajoute la prise en charge de la mémorisation et de la référence à des données désignées. Les mécanismes détaillés dans le présent document vont seulement s'appliquer aux scripts Sieve qui incluent une clause "require" pour l'extension "variables". Les clauses "require" elles mêmes ne sont pas affectées par cette extension.

Les conventions de notation sont celles du paragraphe 1.1 de la [RFC5228], incluant l'utilisation de la [RFC2119] et de la [RFC4234]. La grammaire s'appuie sur celle de la [RFC5228]. Dans ce document, "caractère" signifie un caractère du jeu de caractères codé selon la norme [ISO10646], qui peut consister en plusieurs octets codés selon la [RFC3629], et "variable" est une référence désignée aux données mémorisées ou lues en utilisant les mécanismes de cette extension.

2. Identifiant de capacité

La chaîne de capacités associée à l'extension définie dans ce document est "variables".

3. Interprétation des chaînes

Cette extension change la sémantique de "quoted-string", "multi-line-literal" et "multi-line-dotstuff" donnée dans la [RFC5228] pour permettre l'inclusion de la valeur de "variables".

Quand une chaîne est évaluée, les sous chaînes qui correspondent à "variable-ref" DEVRONT être remplacées par la valeur de "variable-name". Un seul passage à travers la chaîne DEVRA être fait. Les noms de variable sont insensibles à la casse, donc "foo" et "FOO" se réfèrent à la même variable. Les variables inconnues sont remplacées par la chaîne vide.

```
variable-ref = "${" [namespace] variable-name "}"
namespace = identifiant "." *sub-namespace
sub-namespace = variable-name "."
variable-name = num-variable / identifiant
num-variable = 1 *DIGIT
```

Exemples :

"&%\${}" => inchangé, car la chaîne vide est un identifiant illégal.

"\${doh!}" => inchangé, car "!" est illégal dans les identifiants

La variable "company" a la valeur "ACME". Aucune autre variable n'est établie.

"\${full}" => la chaîne vide.

"\${company}" => "ACME"

"\${BAD\${Company}}" => "\${BADACME}"

"\${President, \${Company} Inc.}" => "\${President, ACME Inc.}"

La chaîne étendue DOIT utiliser les valeurs de variable qui sont en cours quand le contrôle atteint la déclaration dont la chaîne fait partie.

Les chaînes où aucune substitution de variable n'a lieu sont appelées des chaînes constantes. De futures extensions pourront spécifier que passer des chaînes non constantes comme arguments à leurs actions ou essais est une erreur.

Les espaces de noms sont destinés à de futures extensions qui rendent l'état interne disponible par des variables. Ces variables DEVRAIENT être mises dans un espace de noms dont le premier composant est le même que sa chaîne de capacités. De telles extensions DEVRAIENT déclarer quelles variables de leur espace de noms sont modifiables avec l'action "set".

Les références aux espaces de noms sans une déclaration "require" préalable pour l'extension pertinente DOIVENT causer une erreur.

Les essais ou actions dans les futures extensions peuvent devoir accéder à la version non étendue de l'argument de chaîne, et, par exemple, faire l'expansion après avoir établi les variables dans son espace de noms. La conception de la mise en œuvre devrait le permettre.

3.1 Caractères de citation et codés

La sémantique de la citation en utilisant la barre oblique inverse n'est pas changée : la citation avec barre oblique inverse (*backslash quoting*) est résolue avant de faire la substitution de variable. De même, le traitement des caractères codés (voir au paragraphe 2.4.2.4 de la [RFC5228]) est effectué avant de faire la substitution de variable, mais après la citation.

Exemples :

"\${fo\o}" => \${foo} => expansion de la variable foo.

"\${fo\o}" => \${fo\o} => identifiant illégal => laissé tel quel.

"\\${foo}" => \${foo} => expansion de la variable foo.

"\\${foo}" => \\${foo} => caractère barre oblique inverse suivi par l'expansion de la variable foo.

Si il est exigé d'inclure une séquence de caractères telle que "\${beep}" telle quelle dans un texte littéral, l'utilisateur peut définir une variable pour circonscrire l'expansion à la chaîne vide.

Exemple :

```
set "dollar" "$";
```

```
set "text" "regarding ${dollar} {beep}";
```

Exemple :

```
require ["encoded-character", "variables"];
set "name" "Ethelbert"
si l'en-tête :contains "Subject" "dear${hex:20 24 7b 4e}ame}" {
# la chaîne d'essai est "dear Ethelbert"
}
```

3.2 Variables de correspondance

Une "variable de correspondance" a un nom consistant en seulement des chiffres décimaux et n'a pas de composant d'espace de noms.

La valeur décimale du nom de la variable de correspondance va être l'indice de la liste des chaînes qui correspondent de la plus récemment évaluée avec succès de type ":matches". La liste est vide si aucune correspondance n'a été trouvée. Note : les zéros en tête supplémentaires sont permis et ignorés.

La liste va contenir une chaîne pour chaque caractère générique ("?" et "**") dans le schéma de correspondance. Chaque chaîne contient la sous chaîne provenant de la valeur de source en laquelle s'étend le caractère générique correspondant, éventuellement la chaîne vide. Les caractères génériques correspondent au minimum (correspondance non avide).

La première chaîne de la liste a l'indice 1. Si l'indice est hors gamme, la chaîne vide va lui être substituée. L'indice 0 contient la partie correspondante de la valeur de source.

L'interpréteur DOIT court-circuiter les essais, c'est-à-dire, ne pas effectuer plus d'essais que nécessaire pour trouver le résultat. L'ordre d'évaluation DOIT être de gauche à droite. Si un essai a deux arguments de liste ou plus, la mise en œuvre est libre de choisir sur lequel itérer en premier.

Une extension qui décrit un nouveau type de correspondance (par exemple, [REGEX]) PEUT spécifier que les variables de correspondance sont réglées par effet collatéral quand le type de correspondance est utilisé dans un script qui a activé l'extension "variables".

Exemple :

```
require ["fileinto", "variables"];
si l'en-tête :matches "List-ID" "*<*@*" {
    fileinto "INBOX.lists.${2}"; stop;
}
```

```
# Imaginons l'en-tête Subject: [acme-users] [fwd] version 1.0 est sortie
si l'en-tête :matches "Subject" "[*] *" {
# ${1} va tenir "acme-users",
# ${2} va tenir "[fwd] version 1.0 est sortie"
    fileinfo "INBOX.lists.${1}"; stop;
}
```

```
# Imaginons l'en-tête To: coyote@ACME.Example.COM
si l'adresse :matches ["To", "Cc"] ["coyote@*.com", "wile@*.com"] {
# ${0} est l'adresse correspondante
# ${1} est toujours la chaîne vide
# ${2} fait partie du nom de domaine ("ACME.Exemple")
    fileinto "INBOX.business.${2}"; stop;
} autrement {
# Le contrôle ne va pas atteindre ce bloc si une correspondance a réussi, donc aucune variables de correspondance n'est e à ce point.
}
```

```
si un de (vrai, adresse :domain :matches "To" "*.com") {
# Le second essai n'est jamais évalué, de sorte qu'il n'y a encore # aucune variable de correspondance établie.
    stop;
}
```

4. Action set

Usage : set [MODIFIER] <nom:chaîne> <valeur: chaîne>

L'action "set" mémorise la valeur spécifiée dans la variable identifiée par le nom. Le nom DOIT être une chaîne constante et se conformer à la syntaxe de "variable-name". Les variables de correspondance ne peuvent pas être établies. Un espace de noms ne peut pas être utilisé si une extension ne permet pas explicitement son usage dans "set". Un nom invalide DOIT être détecté comme une erreur de syntaxe.

Les modificateurs sont appliqués sur une valeur avant qu'elle soit mémorisée dans la variable. Voir les détails au paragraphe suivant.

Les variables sont seulement visibles pour le script en cours. Note : de futures extensions pourraient fournir des règles de script différentes pour les variables.

Les noms de variable ne sont pas sensibles à la casse.

Exemple :

```
set "honorific" "Mr";
set "first_name" "Wile";
set "last_name" "Coyote";
set "vacation" text:
    Cher ${HONORIFIC} ${nom_de_famille},
    Je suis sorti, laissez moi un message après le bip.
:
;
```

"set" n'affecte pas le "keep" implicite. Il est compatible avec toutes les actions définies dans la [RFC5228].

4.1 Modificateurs

Usage : ":lower" / ":upper" / ":lowerfirst" / ":upperfirst" / ":quotewildcard" / ":length"

Les noms de modificateurs sont insensibles à la casse. Les modificateurs inconnus DOIVENT donner une erreur de syntaxe. Plus d'un modificateur peut être spécifié ; dans ce cas, ils sont appliqués conformément à la liste de préséance suivante, le plus grande valeur en premier :

Préséance	Modificateur
40	:lower
	:upper
30	:lowerfirst
	:upperfirst
20	:quotewildcard
10	:length

C'est une erreur d'utiliser deux modificateurs ou plus de la même préséance dans une seule action "set".

Exemples :

```
# La valeur allouée à la variable est imprimée après la flèche
set "a" "IETtReS eMBrouillÉEs"; => "IETtReS eMBrouillÉEs"
set :length "b" "${a}"; => "20"
set :lower "b" "${a}"; => "lettres embrouillées"
set :upperfirst "b" "${a}"; => "IETtReS eMBrouillÉEs"
set :upperfirst :lower "b" "${a}"; => "Lettres embrouillées"
set :quotewildcard "b" "Rock*"; => "Rock\*"
```

4.1.1 Modificateur `":length"`

La valeur est le nombre décimal de caractères dans l'expansion, convertie en une chaîne.

4.1.2 Modificateur `":quotewildcard"`

Ce modificateur ajoute la citation nécessaire pour assurer que le texte expansé va seulement correspondre à une occurrence littérale si il est utilisé comme paramètre à `:matches`. Chaque caractère avec une signification spéciale ("`*`", "`?`", et "`\`") est muni du préfixe "`\`" dans l'expansion.

4.1.3 Modificateurs de casse

Ces modificateurs changent les lettres du texte de majuscule en minuscule ou vice versa. Les caractères autres que "`A`"-"`Z`" et "`a`"-"`z`" de l'US-ASCII restent inchangés.

4.1.3.1 Modificateur `":upper"`

Toutes les lettres minuscules sont converties en leur équivalent majuscule.

4.1.3.2 Modificateur `":lower"`

Toutes les lettres majuscules sont converties en leur équivalent minuscule.

4.1.3.3 Modificateur `":upperfirst"`

Le premier caractère de la chaîne est converti en majuscule si c'est une lettre et réglé en minuscule. Le reste de la chaîne est laissé inchangé.

4.1.3.4 Modificateur `":lowerfirst"`

Le premier caractère de la chaîne est converti en minuscule si c'est une lettre et est réglé en majuscule. Le reste de la chaîne est laissé inchangé.

5. Chaîne d'essai

Usage : `string [MATCH-TYPE] [COMPARATOR] <source: string-list> <key-list: string-list>`

L'essai `"string"` s'évalue à vrai si une des chaînes source correspond à une clé. Le type de correspondance est `":is"` par défaut.

Dans l'essai `"string"`, la source et la liste de clés sont toutes deux prises dans le script, pas dans le message, et la suppression d'espaces NE DOIT PAS être faite si le script ne le demande pas explicitement par un mécanisme futur.

Exemple :

```
set "state" "${state} pending";
si chaîne :matches " ${state} " "*" pending "*" {
# l'essai ci-dessus réussit toujours
}
```

L'extension `"relational"` de la [RFC5231] ajoute un type de correspondance appelé `":count"`. Le compte d'une seule chaîne est 0 si c'est la chaîne vide, ou 1 autrement. Le compte d'une liste de chaînes est la somme des comptes de ses chaînes membres.

6. Limites de mise en œuvre

Une mise en œuvre du présent document DOIT prendre en charge au moins 128 variables distinctes. La longueur supportée des noms de variables DOIT être d'au moins 32 caractères. Chaque variable DOIT être capable de tenir au moins 4000 caractères. Les tentatives d'établissement de variable à une valeur supérieure à ce que la mise en œuvre supporte

DEVRAIENT être rapportées comme une erreur au moment de la compilation, si possible. Si la tentative est découverte pendant le fonctionnement, la valeur DEVRAIT être tronquée, et elle NE DOIT PAS être traitée comme une erreur.

Les variables de correspondance de $\{1\}$ à $\{9\}$ DOIVENT être prises en charge. Les références à des indices supérieurs à ceux que la mise en œuvre supporte DOIVENT être traitées comme des erreurs de syntaxe, qui DEVRAIENT être découvertes au moment de la compilation.

7. Considérations sur la sécurité

Quand des variables de correspondance sont utilisées, et que l'auteur du script n'est pas vigilant, des chaînes peuvent contenir des valeurs arbitraires contrôlées par l'expéditeur du message.

Comme les valeurs mémorisées par "set" qui excèdent les limites de la mise en œuvre sont tronquées en silence, il n'est pas approprié de mémoriser de grandes structures avec des implications de sécurité dans les variables.

L'introduction de variables rend plus facile à écrire la prise de décision évoluée, mais comme aucune construction de bouclage n'est fournie, tous les scripts Sieve vont se terminer de façon ordonnée.

Le filtrage Sieve ne devrait pas être considéré comme une mesure de sécurité contre des messages électroniques hostiles. Sieve est conçu pour faire des essais simples, principalement statiques, et ne convient pas pour être utilisé comme un testeur de pourriels ou de virus, où l'agresseur a comme motivation de faire varier le format du message afin d'éviter les règles de filtrage. Voir aussi la [RFC5235].

8. Considérations relatives à l'IANA

Le gabarit suivant spécifie l'enregistrement par l'IANA des variables d'extension Sieve spécifiées dans ce document :

To : iana@iana.org

Sujet : enregistrement d'une nouvelle extension Sieve.

Nom de capacité : variables

Description : Ajoute la prise en charge de variables au langage de filtrage Sieve.

RFC publiée : RFC 5229

Adresse de contact : liste de diffusion Sieve à [<ietf-mta-filters@imc.org>](mailto:ietf-mta-filters@imc.org)

9. Remerciements

Merci à Cyrus Daboo, Jutta Degener, Ned Freed, Lawrence Greenfield, Jeffrey Hutzelman, Mark E. Mallett, Alexey Melnikov, Peder Stray, et Nigel Swinson pour leurs précieux retours.

10. Références

10.1 Références normatives

[RFC2119] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. (MàJ par [RFC8174](#))

[RFC3629] F. Yergeau, "[UTF-8, un format de transformation](#) de la norme ISO 10646", STD 63, novembre 2003.

[RFC4234] D. Crocker et P. Overell, "[BNF augmenté pour les spécifications de syntaxe](#) : ABNF", octobre 2005. (Remplace [RFC2234](#), remplacée par [RFC5234](#))

[RFC5231] W. Segmuller, B. Leiba, "[Filtrage de messagerie Sieve](#) : extension Relational", janvier 2008. (Remplace [RFC3431](#))(P.S.)

[RFC5228] P. Guenther et autres, "[Sieve : un langage de filtrage](#) de messagerie électronique", janvier 2008. (P.S. ; Remplace [RFC3028](#), MàJ par [RFC5229](#), [5429](#) [9042](#))

10.2 Références pour information

- [ISO-10646] Norme ISO/CEI 10646, "Technologie de l'information – Jeu de caractères universel codé sur plusieurs octets (UCS) - Partie 1 : Architecture et plan multilingue de base", mai 1993, avec amendements.
- [REGEX] Murchison, K., "Sieve Email Filtering -- Regular Expression Extension", Travail en cours, février 2006.
- [RFC5235] C. Daboo, "Filtrage de messagerie Sieve : [extensions Spamtest et Virustest](#)", janvier 2008. (*Remplace RFC3685*) (P.S.)

Adresse de l'auteur

Kjetil T. Homme
University of Oslo
PO Box 1080
0316 Oslo,
Norway

téléphone : +47 9366 0091
mél : kjetilho@ifi.uio.no

Déclaration complète de droits de reproduction

Copyright (C) The IETF Trust (2008).

Le présent document est soumis aux droits, licences et restrictions contenus dans le BCP 78, et à www.rfc-editor.org, et sauf pour ce qui est mentionné ci-après, les auteurs conservent tous leurs droits.

Le présent document et les informations contenues sont fournis sur une base "EN L'ÉTAT" et le contributeur, l'organisation qu'il ou elle représente ou qui le/la finance (s'il en est), la INTERNET SOCIETY et la INTERNET ENGINEERING TASK FORCE déclinent toutes garanties, exprimées ou implicites, y compris mais non limitées à toute garantie que l'utilisation des informations encloses ne viole aucun droit ou aucune garantie implicite de commercialisation ou d'aptitude à un objet particulier.

Propriété intellectuelle

L'IETF ne prend pas position sur la validité et la portée de tout droit de propriété intellectuelle ou autres droits qui pourraient être revendiqués au titre de la mise en œuvre ou l'utilisation de la technologie décrite dans le présent document ou sur la mesure dans laquelle toute licence sur de tels droits pourrait être ou n'être pas disponible ; pas plus qu'elle ne prétend avoir accompli aucun effort pour identifier de tels droits. Les informations sur les procédures de l'ISOC au sujet des droits dans les documents de l'ISOC figurent dans les BCP 78 et BCP 79.

Des copies des dépôts d'IPR faites au secrétariat de l'IETF et toutes assurances de disponibilité de licences, ou le résultat de tentatives faites pour obtenir une licence ou permission générale d'utilisation de tels droits de propriété par ceux qui mettent en œuvre ou utilisent la présente spécification peuvent être obtenues sur le répertoire en ligne des IPR de l'IETF à <http://www.ietf.org/ipr>.

L'IETF invite toute partie intéressée à porter son attention sur tous copyrights, licences ou applications de licence, ou autres droits de propriété qui pourraient couvrir les technologies qui peuvent être nécessaires pour mettre en œuvre la présente norme. Prière d'adresser les informations à l'IETF à ietf-ipr@ietf.org.