

Groupe de travail Réseau
Request for Comments : 5260
 Catégorie : Sur la voie de la normalisation
 Traduction Claude Brière de L'Isle

N. Freed, Sun Microsystems
 juillet 2008

Filtrage de messagerie Sieve : extensions Date et Index

Statut du présent mémoire

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

Résumé

Le présent document décrit les extensions "date" et "index" au langage de filtrage de messagerie Sieve. L'extension "date" donne à Sieve la capacité de vérifier les valeurs de date et d'heure de diverses façons. L'extension "index" donne un moyen de limiter les vérifications d'en-tête et d'adresse à des instances spécifiques de champ d'en-tête quand des champs d'en-tête sont répétés.

Table des matières

1. Introduction.....	1
2. Conventions utilisées dans ce document.....	2
3. Identifiants de capacités.....	2
4. Essai de Date.....	2
4.1 Arguments Zone et Originalzone.....	2
4.2 Argument Date-part.....	3
4.3 Interactions de comparateur avec les arguments Date-part.....	3
4.4 Exemples.....	4
5. Essai Currentdate.....	4
5.1 Exemples.....	4
6. Extension Index.....	5
6.1 Exemple.....	5
7. Considérations pour la sécurité.....	5
8. Considérations relatives à l'IANA.....	6
9. Références.....	6
9.1 Références normatives.....	6
9.2. Références pour information.....	7
Appendice A. Conversions de dates juliennes.....	7
Appendice B. Remerciements.....	8
Adresse de l'auteur.....	8
Déclaration complète de droits de reproduction.....	8

1. Introduction

Sieve [RFC5228] est un langage pour filtrer les messages au moment ou autour du moment de leur livraison finale. Il est conçu pour être mis en œuvre sur un client ou un serveur de messagerie électronique. Il est conçu comme extensible, simple, et indépendant du protocole d'accès, de l'architecture de messagerie, et du système d'exploitation. Il convient au fonctionnement sur un serveur de messagerie où les utilisateurs peuvent n'avoir pas la permission d'exécuter des programmes arbitraires, comme sur les serveurs boîte noire du protocole d'accès au message Internet (IMAP, *Internet Message Access Protocol*) [RFC3501], car il n'a pas de boucle contrôlée par l'utilisateur ou la capacité de faire fonctionner des programmes externes.

L'extension "date" fournit un nouvel essai de date pour extraire et confronter les informations de date/heure des champs d'en-tête structurés. Le concept d'essai de date est similaire à celui d'essai d'adresse spécifié dans la [RFC5228], qui effectue des opérations similaires sur les adresses dans les champs d'en-tête.

L'extension "date" fournit aussi un essai `currentdate` qui opère sur la date et l'heure quand le script Sieve est exécuté.

Certains champs d'en-tête contenant des informations de date/heure, par exemple, `Received:`, se produisent naturellement plus d'une fois dans un seul en-tête. Dans ce cas, il est utile d'être capable de restreindre la vérification de date à un sous-ensemble des champs présents. Par exemple, il peut être utile d'appliquer une vérification de date au dernier (le plus ancien) `Received:`. De plus, il peut aussi être utile d'appliquer des restrictions similaires aux vérifications d'en-tête ou d'adresses spécifiées dans la [RFC5228].

Pour cette raison, la présente spécification définit aussi une extension "index". Cette extension ajoute deux arguments étiquetés supplémentaires `:index` et `:last` aux essais d'en-tête, d'adresse, et de date. Si ils sont présents, ces arguments spécifient quelle occurrence du champ d'en-tête désigné est à vérifier.

2. Conventions utilisées dans ce document

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDÉ", "PEUT", et "FACULTATIF" dans le présent document sont à interpréter comme décrit dans le BCP 14, [RFC2119].

Les termes utilisés pour décrire les divers composants du langage Sieve sont tirés du paragraphe 1.1 de la [RFC5228]. La Section 2 du même document décrit la syntaxe et la sémantique de base du langage Sieve. L'élément syntaxique `date-time` défini en utilisant la notation ABNF de la [RFC5234] dans la [RFC3339] est aussi utilisé ici.

3. Identifiants de capacités

Les chaînes de capacités associées aux deux extensions définies dans ce document sont "date" et "index".

4. Essai de Date

Usage : `date [<":zone" <chaîne time-zone:>> / "":originalzone"] [COMPARATOR] [MATCH-TYPE] <chaîne header-name:> <chaîne date-part:> <liste de chaînes key-list:>`

L'essai `date` confronte les informations de date/heure déduites des en-têtes contenant les valeurs de date-heure de la [RFC2822]. Les informations de date/heure sont extraites de l'en-tête, décalées de la zone horaire spécifiée, et la valeur de la partie date donnée est déterminée. L'essai retourne vrai si la chaîne résultante correspond à une des chaînes spécifiées dans la liste des clés, telle que contrôlée par le comparateur et les mots clés correspondants. L'essai de date retourne faux inconditionnellement si le champ d'en-tête spécifié n'existe pas, si le champ existe mais ne contient pas une spécification de date-heure syntaxiquement valide, si la date-heure n'est pas valide selon les règles du système de calendrier (par exemple, 32 janvier, 29 février dans une année non bissextile) ou si la chaîne résultante ne correspond pas à une valeur de la liste des clés.

Le type de correspondance est par défaut `:is` et le comparateur par défaut est `!;ascii-casemap`.

À la différence des essais d'en-tête et d'adresses, l'essai de date ne peut être appliqué qu'à un seul champ d'en-tête à la fois. Si plusieurs champs d'en-tête avec le même nom sont présents, seul le premier champ trouvé est utilisé. (Noter, cependant, que ce comportement peut être modifié avec l'extension "index" définie ci-dessous.) Ces restrictions simplifient l'essai et gardent claire sa signification.

L'extension "relational" de la [RFC5231] ajoute un type de correspondance appelé `:count`. Le compte d'un essai de date est 1 si le champ spécifié existe et contient une date valide ; 0, autrement.

Les mises en œuvre DOIVENT prendre en charge l'extraction des informations date-heure de la RFC 2822 qui soit mettent à jour le champ d'en-tête entier (par exemple, comme elles le font dans un champ d'en-tête `Date: standard`) soit apparaissent à la fin d'un champ d'en-tête suivant un caractère point-virgule (par exemple, comme elles le font dans un champ d'en-tête `Received: standard`). Les mises en œuvre PEUVENT prendre en charge l'extraction des informations date-heure de la RFC 2822 ou d'autres formats qui apparaissent dans d'autres positions dans le contenu de champ d'en-tête. Dans le cas d'un champ contenant plus d'une valeur de date ou d'heure, la dernière qui apparaît DEVRAIT être utilisée.

4.1 Arguments Zone et Originalzone

L'argument `:originalzone` spécifie que le décalage de zone horaire originellement dans la valeur de date-heure extraite devrait être conservée. L'argument `:zone` spécifie un décalage de zone horaire spécifique auquel la valeur de date-heure doit être décalée avant l'essai. C'est une erreur de spécifier à la fois `:zone` et `:originalzone`.

La valeur de `time-zone` DOIT être un décalage par rapport à l'UTC avec la syntaxe suivante :

```
time-zone = ( "+" / "-" ) 4DIGIT
```

Le "+" ou "-" indique si le `time-of-day` (*heure du jour*) est en avant (c'est-à-dire, à l'est) ou en arrière (c'est-à-dire, à l'ouest) de l'UTC. Les deux premiers chiffres indiquent la différence de nombre d'heures par rapport au temps universel, et les deux derniers chiffres indiquent la différence du nombre de minutes par rapport au temps universel. Noter que ceci est en accord avec le format de la RFC 2822 pour les décalages de zone horaire, pas avec le format de la norme ISO 8601.

Si les deux arguments `:zone` et `:originalzone` sont omis, la zone horaire locale DOIT être utilisée.

4.2 Argument Date-part

L'argument `date-part` spécifie une partie particulière de la valeur de date/heure résultante à confronter à la liste de clés. Les valeurs possibles insensibles à la casse sont :

"year" => l'année, "0000" .. "9999".

"month" => le mois, "01" .. "12".

"day" => le jour, "01" .. "31".

"date" => la date en format "aaaa-mm-jj".

"julian" => le jour julien modifié, c'est-à-dire, la date exprimée comme un nombre entier de jours depuis 00:00 UTC le 17 novembre 1858 (en utilisant le calendrier grégorien). Cela correspond au jour julien régulier moins 2400000,5. Des échantillons de sous programme pour convertir de et vers les dates juliennes modifiées sont donnés à l'Appendice A.

"hour" => l'heure, "00" .. "23".

"minute" => la minute, "00" .. "59".

"second" => la seconde, "00" .. "60".

"time" => l'heure en format "hh:mm:ss".

"iso8601" => la date et l'heure en format ISO 8601 restreint.

"std11" => la date et l'heure dans un format approprié pour l'usage dans un champ d'en-tête `Date:` de la [RFC2822].

"zone" => la zone horaire utilisée. Si l'utilisateur a spécifié une zone horaire avec `:zone`, "zone" va contenir cette valeur.

Si `:originalzone` est spécifié, cette valeur va être la zone originale spécifiée dans la valeur `date-time`. Si aucun argument n'est spécifié, la valeur va être l'heure par défaut du serveur en format décalé "+hhmm" ou "-hhmm". Un décalage de 0 (Zoulou) a toujours un signe positif.

"weekday" => le jour de la semaine exprimé par un entier entre "0" et "6". "0" est dimanche, "1" est lundi, etc.

Le format ISO 8601 restreint est spécifié par la production ABNF `date-time` donnée au paragraphe 5.6 de la [RFC3339], avec les restrictions supplémentaires que les lettres "T" et "Z" DOIVENT être en majuscules, et qu'un décalage de zone horaire de zéro DOIT être représenté par "Z" et non "+00:00".

4.3 Interactions de comparateur avec les arguments Date-part

Tous les comparateurs ne conviennent pas avec tous les arguments `date-part`. En général, les `date-part` peuvent être comparés et vérifiés pour égalité avec `"i;ascii-casemap"` (par défaut) ou `"i;octet"`, mais il y a deux exceptions :

`julian` : c'est un entier, et peut ou non avoir des zéros en tête. À ce titre, `"i;ascii-numeric"` est presque certainement le meilleur comparateur à utiliser avec lui.

`std11` : ceci est fourni comme un moyen d'obtenir les valeurs de `date/time` dans un format approprié pour son inclusion dans des champs d'en-tête de message électronique. La large gamme des syntaxes possibles pour une `date/time` `std11` -- que les mises en œuvre de cette extension sont libres d'utiliser quand elles composent une chaîne `std11` -- rend ce format un choix médiocre pour les comparaisons. Néanmoins, si une comparaison doit être effectuée, c'est insensible à la casse, et donc `"i;ascii-casemap"` doit être utilisé.

"year", "month", "day", "hour", "minute", "second" et "weekday" utilisent tous des représentations de chaîne de longueur fixe d'entiers, et peuvent donc être comparés avec "i;octet", "i;ascii-casemap", et "i;ascii-numeric" avec des résultats équivalents.

"date" et "time" utilisent aussi des représentations de chaîne de longueur fixe d'entiers, et peuvent donc être comparés avec "i;octet" et "i;ascii-casemap" ; cependant, "i;ascii-numeric" ne peut pas être utilisé avec eux, car "i;ascii-numeric" ne permet pas des caractères qui ne sont pas des chiffres.

4.4 Exemples

Le champ Date: peut être vérifié pour voir quand l'envoyeur prétend avoir créé le message et agir en conséquence :

```
require ["date", "relational", "fileinto"];
  if allof(header :is "from" "boss@exemple.com",
    date :value "ge" :originalzone "date" "hour" "09",
    date :value "lt" :originalzone "date" "hour" "17")
  { fileinto "urgent"; }
```

Vérifier le champ initial Received: peut fournir l'indication de quand un message a été réellement reçu par le système local :

```
require ["date", "relational", "fileinto"];
  if anyof(date :is "received" "weekday" "0",
    date :is "received" "weekday" "6")
  { fileinto "weekend"; }
```

5. Essai Currentdate

Usage : currentdate [":zone" <chaîne time-zone:>] [COMPARATOR] [MATCH-TYPE] <chaîne date-part:> <liste de chaînes key-list:>

L'essai currentdate est similaire à l'essai date, sauf qu'il opère sur la date/time courante plutôt que sur une valeur extraite de l'en-tête du message. En particulier, les arguments ":zone" et date-part sont les mêmes que ceux de l'essai date.

Tous les essais currentdate dans un seul script Sieve DOIVENT se référer au même instant durant l'exécution du script.

La valeur :count d'un essai currentdate est toujours 1.

5.1 Exemples

La plus simple utilisation de currentdate est d'avoir une action qui opère seulement à certains moments. Par exemple, un utilisateur pourrait vouloir avoir les messages redirigés sur un autre appareil après les heures de travail et les fins de semaine :

```
require ["date", "relational"];
  if anyof(currentdate :is "weekday" "0",
    currentdate :is "weekday" "6",
    currentdate :value "lt" "hour" "09",
    currentdate :value "ge" "hour" "17")
  { redirect "pager@exemple.com"; }
```

Currentdate peut être utilisé pour établir des réponses de vacance [RFC5230] à l'avance et arrêter automatiquement la génération de réponses :

```
require ["date", "relational", "vacation"];
  if allof(currentdate :value "ge" "date" "2007-06-30",
    currentdate :value "le" "date" "2007-07-07")
```

```
{ vacation :days 7 "Je suis absent durant la première semaine de juillet."; }
```

Currentdate peut aussi être utilisé en conjonction avec l'extension variables pour passer des arguments dépendants de l'heure aux autres essais et actions. Le dialogue Sieve suivant place les messages dans un dossier nommé selon le mois et l'année en cours :

```
require ["date", "variables", "fileinto"];
if currentdate :matches "month" "*" { set "month" "${1}"; }
if currentdate :matches "year" "*" { set "year" "${1}"; }
fileinto "${month}-${year}";
```

Finalement, currentdate peut être utilisé en conjonction avec l'extension editheader pour insérer un champ d'en-tête contenant des informations de date et heure:

```
require ["variables", "date", "editheader"];
if currentdate :matches "std11" "*"
  {addheader "Processing-date" "${0}";}
```

6. Extension Index

L'extension "index", si elle est spécifiée, ajoute les arguments facultatifs :index et :last arguments aux essais d'en-tête, d'adresse, et de date comme suit :

Syntaxe : date [":index" <fieldno: number> [":last"]] [<":zone" <chaîne time-zone:>> / ":originalzone"] [COMPARATOR] [MATCH-TYPE] <chaîne header-name:> <chaîne date-part:> <liste de chaînes key-list:>

Syntaxe : header [":index" <fieldno: number> [":last"]] [COMPARATOR] [MATCH-TYPE] <liste de chaînes header-names:> <liste de chaînes key-list:>

Syntaxe : address [":index" <fieldno: number> [":last"]] [ADDRESS-PART] [COMPARATOR] [MATCH-TYPE] <liste de chaînes header-list:> <liste de chaînes key-list:>

Si :index <fieldno> est spécifié, les tentatives de correspondre à une valeur sont limitées au champ d'en-tête fieldno (commençant à 1, le premier champ d'en-tête nommé). Si :last est aussi spécifié, le compte est à rebours ; 1 note le dernier champ d'en-tête nommé, 2 l'avant dernier, et ainsi de suite. Spécifier :last sans :index est une erreur.

:index compte seulement les champs d'en-tête séparés, pas les occurrences multiples au sein d'un seul champ. En particulier, :index ne peut pas être utilisé pour vérifier une adresse spécifique dans une liste d'adresses contenue dans un seul champ d'en-tête.

L'en-tête et l'adresse permettent tous deux la spécification de plus d'un nom de champ d'en-tête. Si plus d'un nom de champ d'en-tête est spécifié, tous les champs d'en-tête désignés sont comptés dans l'ordre spécifié par la liste d'en-têtes.

6.1 Exemple

La livraison de messagerie peut impliquer plusieurs bonds, résultant en ce que le champ Received: qui contient les informations quand un message est entré pour la première fois dans le domaine administratif local, devient le second champ ou un champ suivant dans le message. Tant que le décalage du champ est cohérent, il peut être vérifié:

Mettre en œuvre la date limite pour le projet Internet en supposant que le second champ Received: spécifie quand le message est entré pour la première fois dans l'infrastructure locale de messagerie électronique

```
.
require ["date", "relational", "index"];
if date :value "gt" :index 2 :zone "-0500" "received"
  "iso8601" "2007-02-26T09:00:00-05:00",
  { redirect "aftercutoff@example.org"; }
```

7. Considérations pour la sécurité

Les facilités définies ici, comme les facilités de la spécification Sieve de base, opèrent sur des informations d'en-tête de message qui peuvent être aisément falsifiées. Noter cependant que certains champs sont par nature plus fiables que d'autres. Par exemple, le champ Date: est normalement inséré par l'envoyeur du message et peut être altéré en tout point. À l'opposé, le champ Received: supérieur est normalement inséré par le système de messagerie local et est donc difficile à falsifier par l'envoyeur ou un intermédiaire.

L'utilisation de la vérification de currentdate rend le comportement du script moins prévisible et plus difficile à analyser. Cela peut avoir des conséquences pour les systèmes qui utilisent l'analyse de script pour essayer et cerner les scripts problématiques.

Toutes les considérations sur la sécurité de la spécification Sieve de base s'appliquent aussi à ces extensions.

8. Considérations relatives à l'IANA

Le gabarit suivant spécifie les enregistrements de l'IANA des deux extensions à Sieve spécifiées dans ce document:

Pour : iana@iana.org

Sujet : enregistrement de nouvelles extensions à Sieve

Nom de capacité : date

Description : l'extension "date" donne à Sieve la capacité de vérifier les valeurs de date et d'heure.

Spécification publiée : RFC 5260

Adresse de contact : liste de discussion Sieve à <ietf-mta-filters@imc.org>

Nom de capacité : index

Description : l'extension "index" fournit un moyen pour limiter les vérifications d'en-tête et d'adresse à des instances spécifiques quand plus d'un champ d'un type donné est présent.

Spécification publiée : RFC 5260

Adresse de contact : liste de discussion Sieve à <ietf-mta-filters@imc.org>

9. Références

9.1 Références normatives

[CALGO199] Tantzen, R., "Algorithm 199: Conversions Between Calendar Date and Julian Day Number", Collected Algorithms from CACM 199.

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

[RFC2822] P. Resnick, "[Format de message Internet](#)", avril 2001. (Remplace la [RFC0822](#), STD 11, Remplacée par [RFC5322](#))

[RFC3339] G. Klyne, C. Newman, "[La date et l'heure sur l'Internet](#) : horodatages", juillet 2002. (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](#))

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

[RFC5234] D. Crocker, P. Overell, "[BNF augmenté pour les spécifications de syntaxe](#) : ABNF", janvier 2008. ([STD0068](#))

9.2. Références pour information

- [RFC3501] M. Crispin, "Protocole d'accès au message Internet - version 4rev1", mars 2003. (P.S. ; MàJ par [RFC4466](#), [4469](#), [4551](#), [5032](#), [5182](#), [7817](#), [8314](#), [8437](#), [8474](#) ; remplacée par la RFC9051
- [RFC5230] T. Showalter, N. Freed, éd., "[Filtrage de messagerie Sieve](#) : extension Vacation", janvier 2008. (P.S. ; MàJ par [RFC8580](#).)

Appendice A. Conversions de dates juliennes

Le sous programme en langage C suivant montre comment traduire les informations de jour/mois/année en dates juliennes modifiées. Ces sous programmes sont des traductions directes des sous programmes Algol spécifiés dans l'algorithme CACM 199 [CALGO199].

Étant donné le jour, mois, et an, jday retourne la date julienne modifiée.

```
int jday(int year, int month, int day)
{
    int j, c, ya;
    if (month > 2)
        month -= 3;
    else
    {
        month += 9;
        year--;
    }
    c = year / 100;
    ya = year - c * 100;
    return (c * 146097 / 4 + ya * 1461 / 4 + (month * 153 + 2) / 5 + day + 1721119);
}
```

Étant donné j, la date julienne modifiée, jdate retourne le jour, mois, et année.

```
void jdate(int j, int *year, int *month, int *day)
{
    int y, m, d;
    j -= 1721119;
    y = (j * 4 - 1) / 146097;
    j = j * 4 - y * 146097 - 1;
    d = j / 4;
    j = (d * 4 + 3) / 1461;
    d = d * 4 - j * 1461 + 3;
    d = (d + 4) / 4;
    m = (d * 5 - 3) / 153;
    d = d * 5 - m * 153 - 3;
    *day = (d + 5) / 5;
    *year = y * 100 + j;
    if (m < 10)
        *month = m + 3;
    else
    {
        *month = m - 9;
        *year += 1;
    }
}
```

Appendice B. Remerciements

Dave Cridland a contribué au texte qui décrit les comparateurs appropriés à utiliser avec les différentes parties de date. Cyrus Daboo, Frank Ellerman, Alexey Melnikov, Chris Newman, Dilyan Palauzov, et Aaron Stone ont fourni des suggestions et corrections utiles.

Adresse de l'auteur

Ned Freed
Sun Microsystems
800 Royal Oaks
Monrovia, CA 91016-6347
USA
téléphone : +1 909 457 4293
mél : ned.freed@mrochek.com

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 sauf pour ce qui est mentionné ci-après, les auteurs conservent tous leurs droits.

Le présent document et les informations y contenues sont fournies 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, le IETF TRUST 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 ci-encloses ne violent 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 pourrait 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 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.