

# RFC 5804 : A Protocol for Remotely Managing Sieve Scripts

Stéphane Bortzmeyer  
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 14 juillet 2010

Date de publication du RFC : Juillet 2010

<https://www.bortzmeyer.org/5804.html>

---

Le langage de filtrage du courrier Sieve permet d'écrire des scripts décrivant les actions à effectuer automatiquement en fonction du contenu d'un message électronique. Ces scripts peuvent évidemment être édités avec vi directement sur le serveur mais, très souvent, les utilisateurs de Sieve n'ont pas la possibilité de se connecter directement audit serveur. Ce RFC normalise donc un protocole, **ManageSieve**, qui permet de mettre à jour des scripts Sieve sur le serveur.

Comme résumé en sections 1.2 et 1.3, ManageSieve est un protocole texte, orienté ligne, analogue à de nombreux autres protocoles IETF comme IMAP. Le client ManageSieve se connecte au serveur, s'authentifie, envoie des commandes et le serveur répond par des codes (section 1.4) indiquant le succès ou l'échec de la commande.

Voici un exemple de session où les lignes préfixées par C : sont envoyées par le client et celles préfixées par S : sont les réponses du serveur :

```
S: "IMPLEMENTATION" "Example ManageSieved v001"
S: "SIEVE" "fileinto vacation"
S: "SASL" "PLAIN"
S: "STARTTLS"
S: "VERSION" "1.0"
S: "LANGUAGE" "fr"
S: OK
C: Authenticate "PLAIN" "QJlrweAPyo6Q1T9xy"
S: OK
C: Putscrip "mysievescript" {110+}
C: require ["fileinto"];
C:
C: if envelope :contains "to" "tmartin+sent" {
C:   fileinto "INBOX.sent";
C: }
S: OK
```

L'utilisateur s'est authentifié, puis a déposé un script Sieve nommé `mysievescript`.

Une notion importante du protocole ManageSieve est celle de **script actif** (section 1.5). ManageSieve permet de déposer plusieurs scripts sur le serveur, chacun identifié par un nom (les noms sont décrits en section 1.7, ils peuvent être pratiquement n'importe quelle chaîne Unicode) mais un seul est script actif (est réellement exécuté lors de la délivrance du courrier). La commande `SETACTIVE` (section 2.8) sert à rendre un script actif.

Comme la plupart des protocoles, ManageSieve impose un certain nombre de fonctions qui **doivent** être mises en œuvre, et d'autres qui sont optionnelles (section 1.8). Un serveur ManageSieve annonce ces **capacités** à l'établissement de la connexion et en réponse à la commande `CAPABILITY` (section 2.4). Parmi elles, la possibilité de protéger la session par TLS (obligatoire), la liste des extensions Sieve acceptées (obligatoire), la langue utilisée pour les messages d'erreur (optionnelle), etc <<https://www.iana.org/assignments/managesieve/managesieve.xhtml>>.

ManageSieve fonctionne sur TCP (section 1.9), le port par défaut étant 4190 (attention, pendant longtemps, c'était le 2000 et plusieurs logiciels ont encore cela par défaut). L'adresse IP du serveur est trouvée en résolvant d'abord les enregistrements `SRV _sieve._tcp.$DOMAIN` et, si le SRV n'existe pas, en essayant directement une adresse liée au nom de domaine.

La liste complète des commandes possibles figure en section 2. Deux exemples :

- `PUTSCRIPT` (section 2.6) permet de déposer un script en lui donnant un nom. À noter que le serveur **doit** tester la validité syntaxique du script avant de l'enregistrer.
- `LISTSCRIPTS` (section 2.7) qui donne la liste des scripts Sieve déposés, en indiquant lequel est actif.

On peut désigner un serveur ManageSieve, ou bien un script sur un serveur, avec un URL (section 3). Un exemple est `sieve://example.net/durand/mon-script` qui identifie le script `mon-script` de l'utilisateur `durand` sur le serveur ManageSieve du domaine `example.net`.

Bien que le RFC vienne de paraître, ManageSieve est un ancien protocole et est déjà mis en œuvre, côté client dans SquirrelMail <[http://www.squirrelmail.org/plugin\\_view.php?id=73](http://www.squirrelmail.org/plugin_view.php?id=73)>, Emacs <<http://josefsson.org/sieve/>>, un simple script Perl (en ligne sur <https://www.bortzmeyer.org/files/sieve.pl>), les programmes `sieve-connect` et `sieveshell` de Cyrus et, côté serveur, dans Archiveopteryx <<http://www.archiveopteryx.org/sieve/>>, Dovecot <<http://wiki.dovecot.org/ManageSieve>> (voir aussi PigeonHole <<http://pigeonhole.dovecot.org/>>), Cyrus (programme `timsieved`)... Il existe aussi un serveur semi-autonome en Python, `pysieved` <<http://gitorious.org/pysieved>>. Sur la question générale du filtrage du courrier, on peut consulter l'article « Filtres côté serveur » <[http://casys.crevetor.org/index.php/Filtres\\_ct\\_serveur](http://casys.crevetor.org/index.php/Filtres_ct_serveur)> ».