

---

# rsync

## Outil de copie de fichier

rsync est un outils de copie de fichier versatile. Il peut copier localement, depuis/vers un autre autre via un shell distant, ou vers/depuis un service rsync distant. Il offre un grand nombre d'options qui contrôle tous les aspects de son fonctionnement, et permet de spécifier de manière très flexible le jeu de fichiers à copier. Il à un algorithme de transfert delta, qui réduit la quantité de données envoyées sur le réseau en envoyant seulement les différences entre les fichiers sources et les fichiers existants dans la destination. rsync est largement utilisé pour les sauvegardes et le mirroring et comme commande de copie améliorée.

rsync trouve les fichiers qu'il doit transférer en utilisant un algorithme vérification rapide (par défaut) qui recherche les fichier qui ont changé en taille ou de mtime. Tout changement dans les autres attributs préservés (comme requis par les options) sont fait dans le fichier de destination directement quand la vérification rapide indique que les données du fichiers n'ont pas besoin d'être mis à jours.

rsync copies les fichier depuis ou vers un hôte distant, ou localement. Il y a 2 manières différentes pour rsync pour contacter un système distant : utiliser un shell distant (tel que ssh ou rsh), ou contacter un service rsync directement via TCP. le transport par shell distant est utilisé quand le chemin source ou de destination contient un séparateur " :" après la spécification de l'hôte. Contacter un service rsync se fait en spécifiant le séparateur ' : :' après la spécification de l'hôte, ou quand une url rsync :// est spécifiée.

Un cas spéciale, si un argument de source simple est spécifié dans destination, les fichiers sont listés dans un format de sortie similaire à 'ls -l'.

Comme attendu, si ni le chemin source ni de destination ne spécifie d'hôte distant, la copie se produit localement.

rsync réfère au côté local comme le 'client' et le côté distant comme le 'serveur', à ne pas confondre avec le service rsync, qui est toujours un serveur, mais un serveur qui peut être soit un service ou un processus lancé par un shell distant.

## Utilisation

Il est possible d'utiliser rsync de la même manière que rcp, à l'exception que rsync n'accepte qu'un seul hôte distant.

La meilleur manière d'expliquer la syntaxe est avec certains exemples :

**rsync -t \*.c foo :src/**

Cela transfère tous les fichiers correspondant au motif \*.c du répertoire courant vers un répertoire src sur la machine foo. Si un des fichiers existe déjà sur le système distant, le protocole de mise à jours distant de rsync est utilisé pour mettre à jours le fichier en envoyant seulement les données différente. Noter que l'expansion du wildcard sur la ligne de commande en une liste de fichiers est gérée par le shell avant de lancer rsync.

**rsync -avz foo :src/bar /data/tmp**

Transfère récursivement tous les fichiers du répertoire src/bar dans la machine foo dans le répertoire local /data/tmp/bar. Les fichiers sont transférés

## OPTIONS

**-v, -verbose** Augmente la quantité d'informations données durant le transfert. Peut être spécifié plusieurs fois.

**-info=FLAGS** Permet de spécifier plus finement les informations à afficher. Utiliser **-info=help** pour voir les flags disponibles. les flags sont séparés par des ","

**-debug=FLAGS** Idem mais avec des informations plus verbeuses.

**-msg2stderr** Envoie sa sortie sur stderr au lieu de stdout. Principalement pour debug.

- 
- q, -quiet** Mode silencieux, utile avec cron
  - no-motd** Affecte les informations qui sont affichés au démarrage du transfert du service. Il supprime le mot du jours et la liste des modules.
  - I, -ignore-times** Normalement rsync saute les fichiers qui existent déjà avec la même taille et le même mtime. Cette option désactive le mode vérification rapide, forçant tous les fichiers à être mis à jours.
  - size-only** Modifie l'algorithme de vérification rapide pour trouver les fichiers qui doivent être transférés. Seuls les fichiers avec une taille différente sont mis à jours. Utile pour utiliser rsync avec un système de mirroring qui ne préserve pas les timestamps.
  - modify-window** En comparant 2 timestamps, rsync traite les timestamps comme étant égaux s'ils diffèrent par moins que la valeur modify-window. C'est par défaut 0. En particulier, en transférant depuis ou vers un système de fichier FAT qui a une précision à 2 secondes, -modify-window=1 est utile.
  - c, -checksum** Change la manière dont rsync vérifie si les fichiers ont été changés et ont besoin d'être transférés. Sans cette options, rsync utilise la vérification rapide. Ce option compare un checksum 128-bits pour chaque fichier. La génération des checksums signifie que le client et le serveur vont consommer plus de lecture disque, ce qui peut ralentir les choses significativement.
  - a, -archive** C'est équivalent à **-rlptgoD**. C'est une manière rapide de dire que vous souhaitez une récursion et préserver tout. La seule exception par rapport à l'équivalence et quand **-files-from** est spécifié, auquel cas **-r** n'est pas induit. Noter que -a ne préserve pas les liens hard, utiliser -H.
  - no-OPTION** Permet de désactiver une option en préfixant le nom de l'option avec **-no**. Toutes les options ne peut pas être préfixé de cette manière, seules les options qui sont sous-entendus par d'autres options (ex : -no-D, -no-perms) ou ont des valeurs par défaut en fonction des circonstances (ex : -no-whole-file, -no-blocking-io, -no-dirs). Note : l'ordre des options est importante
  - r, -recursive** Dit à rsync de copier les répertoires récursivement. La récursivité est incrémentale, le transfert commence avant que tous les répertoires aient été scannés. certaines options nécessite un scan complet et désactivent la récursivité incrémentale.
  - R, -relative** Utilise les chemins relatifs. Le chemin complet spécifié sur la ligne de commande est envoyé au serveur au lieu de simplement envoyer la dernière partie des noms de fichiers.
  - no-implied-dirs** Affecte le comportement de l'option -relative. Si spécifié, les attributs des répertoires implicites des noms sources ne sont pas inclus dans le transfert. Cela signifie que les éléments de chemins sur la destination ne sont pas changé, et les répertoires créés le sont avec des attributs par défaut.
  - b, -backup** Avec cette option, des fichiers de destination pré-existant sont renommé quand le fichier est transféré ou supprimé.
  - backup-dir=DIR** Avec -backup, indique où stocker les backup. Peut être utilisé pour les backups incrémentals.
  - suffix-SUFFIX** Permet de changer le suffixe backup par défaut pour l'option -backup. Défaut : "~" ou vis si -backup-dir a été spécifié
  - u, -update** Force rsync à ignorer tout fichier existant dans la destination
  - inplace** Change la méthode de transfert d'un fichier quand les données sont mis à jours. Au lieu de créer une nouvelle copie du fichier et en le déplaçant une fois complété, rsync écrit directement dans le fichier de destination
  - append** Met à jours un fichier en ajoutant les données à la fin du fichier, ce qui suppose que les données déjà existante dans la destination est identique avec le début du fichier.
  - append-verify** Similaire à -append, mais les données existant sont inclus dans le checksum de vérification.
  - d, -dirs** Indique à l'émetteur d'inclure tous les répertoire rencontrés. À la différence de -recursive, le contenu des répertoires ne sont pas copiés sauf si le nom du répertoire est "." ou se termine par "/". -recursive a précédence.
  - l, -links** Quand les liens sont rencontrés, recréé les liens dans la destination
  - L, -copy-links** Quand des liens sont rencontrés, l'élément pointé est copié au lieu du lien.
  - copy-unsafe-links** Copie la référence des liens symboliques en dehors de l'arborescence copiée.
  - safe-links** Ignore les liens symboliques pointant en dehors de l'arborescence copiée.
  - munge-links** Modifie tous les liens symbolique reçu pour les rendre inutilisable, mais récupérable, ou réactive les liens stockés dans un état munge. rsync désactive les liens en les préfixant avec la chaîne /rsync-munged/. Celà empêche d'utiliser les liens tant que ce répertoire n'existe pas.
  - k, -copy-dirlinks** Traite les liens vers un répertoire comme si c'est un vrai répertoire
  - K, -keep-dirlinks** idem à -k, mais seulement s'il matche un vrai répertoire de l'émetteur.
  - H, -hard-links** Recherche les liens hard dans la source et les lie ensemble aux fichiers correspondant dans la destination.
  - p, -perms** Copie les permissions
  - E, -executability** Préserve la permission 'x' des fichiers réguliers quand -perms n'est pas activé.
  - A, -acls** Met à jours les ACL. Implique également -perms

- 
- X**, **-xattrs** met à jours les attributs étendus
  - chmod** Applique un ou plusieurs modes de permission aux fichiers transférés.
  - o**, **-owner** Définis le propriétaire du fichier de destination
  - g**, **-group** Définis le groupe propriétaire du fichier de destination
  - devices** Transfert les fichiers caractère et block
  - specials** transfère les fichiers spéciaux tels que les fichiers sockets et fifo
  - D** = **-devices** **-spécials**
  - t**, **-times** Transfert les horodatages de modification des fichiers
  - O**, **-omit-dir-times** Omet les répertoire avec **-times**. Utile avec NFS sur des répertoires partagés
  - J**, **-omit-link-times** Omet les liens symboliques en préservant les dates de modifications (**-times**)
  - super** Tente les activités root même si le receveur n'est par root : **-owner**, **-groups** et **-devices**.
  - fake-super** Simule les activités root en sauvant/restaurant les attributs privilégiés via des attributs étendus spéciaux attachés à chaque fichier (si nécessaire). Gère également **-acls** et **-xattrs** s'ils sont spécifiés
  - S**, **-sparse** Tente de gérer les fichiers sparsés efficacement pour qu'ils prennent moins d'espace sur la destination. Est en conflit avec **-inplace**
  - preallocate** Indique au receveur d'allouer chaque fichier de destination avant d'écrire les données. rsync utilise le support de préallocation du système de fichier via `fallocate(2)`. Avec cette options les gros fichier peuvent ne pas être contigus.
  - n**, **-dry-run** Effectue une simulation qui ne change rien. Utile avec **-verbose** et/ou **-itemize-change**
  - W**, **-whole-file** Désactive l'algorithme de transfert delta de rsync. Peut être plus rapide si la bande passante entre la source et la destination est supérieure à la bande passante du disque.
  - checksum-choice=STR** Change l'algorithme de checksum. (auto, md4, md5, none)
  - x**, **-one-file-system** Ne traverse pas les systèmes de fichier lors des récursion. Si cette option est répétée, rsync omet tous les répertoires de point de montage qu'il rencontre
  - existing**, **-ignore-non-existing** Saute la création de fichiers et répertoires qui n'existent pas dans la destination. Combinée avec **-ignore-existing**, aucun fichier n'est mis à jours.
  - ignore-existing** Met à jours les fichiers qui existent déjà dans la destination.
  - remove-source-files** Indique à rsync de supprimer les fichiers envoyés qui font partie du transfert et ont été dupliqués avec succès
  - delete** Supprime les fichiers dans le receveur (ceux qui n'ont pas été envoyés), mais uniquement pour les répertoires qui sont synchronisés
  - delete-before** Demande que la suppression de fichiers dans le receveur soit fait avant que le transfert commence.
  - delete-during**, **-del** Demande que la suppression de fichiers dans le receveur soit fait incrémentalement durant le transfert
  - delete-delay** Demande que la suppression de fichiers dans le receveur soit calculé durant le transfert, puis supprimé à la fin du transfert
  - delete-after** Demande que la suppression de fichiers dans le receveur soient fait une fois le transfert complété.
  - delete-excluded** En plus de supprimer les fichiers dans le receveur qui ne sont pas dans l'émetteur, indique de supprimer également tous fichier dans le receveur qui sont exclus
  - ignore-missing-args** Quand rsync traite d'abord les fichiers sources demandés explicitement (ex **-files-from**), si le fichier n'est pas trouvé, ne génère pas d'erreur et ne tente pas de transférer le fichier.
  - delete-missing-args** Idem, mais chaque argument manquant devient une requête de suppression
  - ignore-errors** **-delete** supprime les fichier quand il y a des erreurs d'entrées/sorties
  - force** Supprime un répertoire non-vide quand il est remplacé par un non-répertoire.
  - max-delete=NUM** Indique de ne pas supprimer plus de NUM fichiers/répertoires. Au delà, toutes les suppressions sont ignorées. rsync retourne avec un code d'erreur 25
  - max-size=SIZE** Évite les transferts de fichiers supérieurs à la taille spécifiée.
  - min-size=SIZE** Évite les transferts de fichiers inférieurs à la taille spécifiée.
  - B**, **-block-size=BLOCKSIZE** Force la taille de block utilisé dans l'algorithme de transfert delta à une taille fixe.
  - e**, **-rsh=COMMAND** Permet de choisir un shell distant alternatif à utiliser pour la communication.
  - rsync-path=PROGRAM** Spécifie le programme à lancer sur la machine distante pour démarrer rsync.
  - M**, **-remote-option=OPTION** Utilisé pour limiter certains effets sur le transfert, comme **-log-file** et **-fake-super**

- C, **-cvs-exclude** Raccourci utile pour exclure une grande plage de fichier. la liste d'exclusion est : RCS SCCS CVS CVS.adm RCSLOG cvslog.\* tags TAGS .make.state .nse\_depinfo \*~#\* .#\* ,\* \_\$\* \$\$ \*.old \*.bak \*.BAK \*.orig \*.rej .del-\* \*.a \*.olb \*.o \*.obj \*.so \*.exe \*.Z \*.elc \*.ln core .svn/ .git/ .hg/ .bzr/
- f, **-filter=RULE** Ajoute une règle d'exclusion de fichier
- F Raccourci pour ajouter 2 **-filter**, la première fois est un raccourci à **-filter='dir-merge /rsync-filter'**, qui indique de rechercher les fichiers **rsync-filter** dans les répertoire et d'utiliser leurs règles.
- exclude=PATTERN** Forme simplifiée de l'option **-filter**
- exclude-from=FILE** Cette option est liée à **-exclude**, mais spécifie un fichier qui contient des motifs d'exclusion.
- include=PATTERN** Forme simplifiée de **-filter** qui crée une règle d'inclusion
- include-from=FILE** Liée à **-include**, mais spécifie un fichier qui contient des motifs d'inclusion
- files-from=FILE** Spécifie la liste exacte de fichiers à transférer. Il personnalise également le comportement de **rsync** :
  - R, **-relative** Implicite, et préserve les informations de chemin qui sont spécifiés pour chaque éléments dans le fichier.
  - d, **-dirs** Implicite, qui crée les répertoires spécifiés dans la liste dans la destination
  - a, **-archive** n'implique pas **-recursive**
- O, **-fromO** Les règles/noms de fichiers lus depuis un fichier sont terminés par un caractère null (\0). Affecte **-exclude-from**, **-include-from**, **-files-from**, et tous fichiers fusionnés spécifiés dans une règle **-filter**
- s, **-protect-args** Envoie tous les noms de fichier et les options au **rsync** distant sans permettre au shell distant de les interpréter.
- T, **-temp-dir=DIR** Utiliser le répertoire comme répertoire scratch en créant des copies temporaires des fichiers transférés. Le comportement par défaut est de créer chaque fichier temporaire dans le même répertoire que le fichier de destination associé
- y, **fuzzy** Indique à **rsync** qu'il doit rechercher un fichier de base pour tout fichier de destination manquant. l'algorithme actuel recherche dans le même répertoire que le fichier de destination un fichier qui a une taille et un mtime identique, ou un fichier de même nom. Si trouvé, **rsync** utilise le fichier pour accélérer le transfert. Si cette option est répétée, le fuzzy scan également tous les répertoires de destination correspondant qui sont spécifié via **-compare-dest**, **-copy-dest** ou **-link-dest**
- compare-dest=DIR** Utilise le répertoire dans la destination comme hiérarchie additionnelle pour comparer les fichiers de destination. Si un fichier est trouvé dans ce répertoire qui est identique au fichier à envoyer, le fichier n'est pas transféré
- copy-dest=DIR** Identique à **-compare-dest**, mais **rsync** copie également les fichiers inchangés trouvés dans ce répertoire. Peut être spécifié plusieurs fois
- link-dest=DIR** identique à **-copy-dest**, mais les fichiers inchangés sont hardlinked depuis ce répertoire .
- z, **-compress** Comprime les données envoyée
- compress-level=NUM** Niveau de compression
- skip-compress=LIST** Liste de suffixe de fichier qui ne sont pas compressés. La liste par défaut est : 7z ace avi bz2 deb gpg gz iso jpeg jpg lz lzma lzo mov mp3 mp4 ogg png rar rpm rzip tbz tgz tlz txz xz z zip
- numeric-ids** Transfert les UID/GID au lieu d'utiliser les noms d'utilisateur/groupe.
- usermap=STRING, -groupmap=string** Spécifie les utilisateurs et groupe qui doivent être mappés à d'autres valeurs dans le receveur. Le chaîne consiste d'une ou plusieurs paires de FROM :TO.
- chown=USER :GROUP** Force tous les fichiers à être possédés par l'utilisateur et groupe spécifiés
- timeout=TIMEOUT** Permet de définir un timeout d'E/S maximum, en secondes. Si aucune données n'est transférée durant ce laps de temps, **rsync** quitte.
- contimeout** Délais d'attente pour une connexion avant de quitter
- address** Spécifie une adresse IP de connexion
- port=PORT** Port TCP pour la connexion. Défaut : 873
- sockopts** Fournis des options de socket. voir **setsockopt(2)**
- blocking-io** Utilise les E/S bloquant en lançant un shell distant. Si le shell distant est **rsh** ou **remsh**, **rsync** l'utilise par défaut
- outbuf=MODE** Définis le mode de buffering par défaut. Peut être **None**, **Line**, ou **Block**
- i, **-itemize-changes** Demande une simple liste itemisée de changement qui sont faits à chaque fichier, incluant les changements d'attributs. Identique à **-out-format='%i %n%L'**.
- out-format=FORMAT** Spécifie ce que le client **rsync** affiche à l'utilisateur sur une base par mise à jours. Défaut : "%n%L" si **-info** ou **-v** est spécifié.
- log-file=FILE** log les opérations dans un fichier

---

**-log-file-format=FORMAT** Format des messages loggés Défaut : '%i %n%L'

**-stats** Affiche un jeu de statistiques durant le transfert. Cette option est équivalente à **-info=stats2** ou **-info=stats3**.

**-8, -8-bit-output** laisse les caractères 8-bits non échappés dans la sortie au lieu de les tester pour voir s'ils sont valides dans la locale courante

**-h, -human-readable** Affiche les nombres au format human-readable

**-partial** par défaut, rsync supprime tout fichier partiellement transféré si le transfert a été interrompu. cette option permet de conserver le fichier partiel pour accélérer les transferts suivants

**-partial-dir=DIR** Spécifie un répertoire utilisé pour maintenir les données partielles

**-delay-updates** Place le fichier temporaire de chaque fichier mis à jours dans un répertoire jusqu'à la fin du transfert, puis tous les fichiers sont renommés. Cela rend les mises à jours un peu plus atomiques mais utilise plus de mémoire

**-m, -prune-empty-dirs** Indique au receveur de gérer les répertoires vides depuis la liste des fichiers, incluant les répertoires imbriqués qui ont des enfants non-répertoire. C'est utile pour éviter la création d'un grand nombre de répertoires inutiles

**-progress** Affiche la progression du transfert

**-P** Équivalent à **-partial -progress**

**-password-file=FILE** Permet de fournir un mot de passe pour accéder au service rsync via un fichier ou via l'entrée standard avec '-'

**-list-only** Liste les fichiers sources au lieu de les transférer

**-bwlimit=RATE** Limite le taux de transfert

**-write-batch=FILE** Enregistre un fichier qui peut être appliqué ultérieurement à une autre destination identique avec **-read-batch**.

**-only-write-batch=FILE** Similaire à **-write-batch**, excepté qu'aucune mise à jours n'est faite dans le système de destination en créant le batch

**-read-batch=FILE** Applique tous les changements stockés dans le fichier précédemment généré par **-write-batch**

**-protocol=NUM** Permet de forcer une ancienne version de protocole. Utile avec **-write-batch**

**-iconv=CONVERT\_SPEC** Convertit les noms de fichiers entre les jeux de caractères. '.' indique à rsync de regarder le jeu de caractères par défaut

**-4, -ipv4, -6, -ipv6** Socket à utiliser

**-checksum-seed=NUM** Définis le checksum d'envoi. Ce checksum de 4 octets est inclus dans chaque block et chaque calcul MD4 (MD5 n'utilise pas de seed).

## Options pour le mode service

**-daemon** Lance rsync en mode service. Le service devient accessible en utilisant `host : :module` ou `rsync ://host/module/`

**-address** IP d'écoute du service

**-bwlimit=RATE** Permet de spécifier le taux de transfert maximum pour les données que le service envoie

**-config=FILE** Spécifie le fichier de configuration

**-M, -dparam=OVERRIDE** Définis un paramètre de configuration spécifique au service

**-no-detach** Ne détache pas le service en tâche de fond

**-port=PORT** Spécifie le port TCP d'écoute

**-log-file=FILE** Remplace le paramètre log file dans la configuration

**-log-file-format=FORMAT** Remplace le paramètre log format dans la configuration

**-sockopts** Remplace le paramètre socket options dans la configuration

**-v, -verbose** mode verbeux

**-4, -ipv4, -6, -ipv6** Spécifie la famille d'adresse d'écoute

## Règles de filtrage

---

Les règles de filtrage permet une sélection flexible des fichiers à transférer et à exclure. Les règles spécifient soit directement des motifs d'inclusion/exclusion pour spécifie une manière d'acquérir plus de motifs (par ex en lisant un fichier)

Vu que la liste des fichiers/répertoires est construite, rsync vérifie chaque nom à transférer avec la liste, et le premier match est acté. rsync construit une liste ordonnée de règles de filtrage tel que spécifié dans la ligne de commande. Les règles de filtrage ont la syntaxe suivante :

```
RULE [PATTERN_OR_FILENAME]
```

```
RULE,MODIFIERS [PATTERN_OR_FILENAME]
```

Le motif ou nom de fichier qui suit, si présent, doit venir après soit un simple espace, ou un ' '. Les préfixes de règles disponibles sont :

**exclude, -** spécifie un motif d'exclusion

**include, +** Spécifie un motif d'inclusion

**merge, .** Spécifie un fichier où liste plus de règles

**dir-merge, :** Spécifie un fichier par répertoire

**hide, H** Spécifie un motif pour cacher des fichiers du transfert

**show, S** Spécifie un motif pour afficher des fichiers

**protect, P** Spécifie un motif pour protéger les fichiers de la suppression

**risk, R** Ces fichier ne sont pas protégés

**clear,!** efface la liste d'inclusion/exclusion courante, n'a pas d'argument.

Noter que les options `--include/--exclude` ne permettent pas tous les préfixes, ils autorisent seulement les motifs d'inclusion/exclusion, et `!` pour effacer la liste. Si un motif ne commence pas avec `"- "` ou `"+ "`, la règle est interprétée comme si `"+ "` (pour une option include), ou `"- "` (pour une option exclude). Une option `--filter` doit toujours contenir un nom de règle court ou long au début de la règle.

## Règles de motif d'exclusion/inclusion

Pour inclure et exclure des fichier, utiliser `"+"`, `"-"`, etc. Les règles spécifient un motif qui est matché avec les noms des fichiers qui doivent être transférés. Ces motifs peuvent prendre plusieurs forme :

- Si le motif commence avec un `'/'`, il est attaché à un emplacement particulier dans la hiérarchie des fichiers, sinon il est matché avec la fin d'un nom de chemin. C'est similaire à un `'^'` de début dans les expressions régulières. donc `/foo` matche un nom `foo` soit à la racine du transfert ou dans le répertoire dans le cas d'un merge-file.
- Si le motif se termine avec un `/`, il ne matche qu'un répertoire, pas un fichier régulier, ni lien ni périphérique.
- rsync choisi entre faire un simple match et un match wildcard en vérifiant si le motif contient un des caractères wildcard `'*'`, `'?'`, et `'['`
- un `'*'` matche tous composant de chemin, mais il s'arrête aux `'/'`
- `addentry articles autoadd autofind autoprod createalpha createbeta createdb createprod findentry fullpowa generator.php genhtml genman genmd gentex html images insert man md pdf regen setfor setfor2 sql temp tex threads ToDo Match tout`, incluant les `'/'`
- `?` Matche tout caractère excepté `'/'`
- `[` introduit une classe de caractères
- Dans un motif wildcard, un `'\'` permet d'échapper un caractère wildcard.
- Si le motif contient un `'/'` (sans compter le `'/'` final) ou un `'***'`, il est matché avec le chemin complet. Si le motif ne contient pas de `'/'` ou `'***'`, il est matché seulement avec le composant final du nom de fichier.
- un `'dir_name/***'` final match le répertoire et tout ce qu'il contient.

Noter que, en utilisant `--recursive`, qui est implicite avec `-a`, tous les sous-répertoires de tous les chemins sont visités de gauche à droite, chaque répertoire ayant une chance d'être exclus avant son contenu. De cette manière les motifs d'inclusion/exclusion sont appliqués récursivement au nom de fichier de chaque nœud dans le système de fichier. Les motifs d'exclusion court-circuitent l'étape de parcours du répertoire.

---

Par exemple, pour inclure `/foo/bar/baz`, les répertoires `/foo` et `/foo/bar` ne doivent pas être exclus. Exclure un de ces parent empêche d'examiner son contenu. Le concept d'exclusion est particulièrement important en utilisant un `'**'` final, par exemple, ceci ne fonctionne pas :

```
+ /some/path/this-file-will-not-be-found
+ /file-is-included
- *
```

Une solution est de demander à tous les répertoires dans la hiérarchie d'être inclus en utilisant une simple règle `"+ */"`, et utiliser l'option `-prune-empty-dirs`. Une autre solution est d'ajouter des règles d'inclusion spécifiques pour tous les répertoires parents qui doivent être visités :

```
+ /some/
+ /some/path/
+ /some/path/this-file-is-found
+ /file-also-included
- *
```

`'- *.o'` Exclut tous les nom matchant `*.o`

`'- /foo'` Exclut un fichier ou répertoire nommé `foo` dans le répertoire racine du transfert

`'- foo/'` Exclut tout répertoire nommé `foo`

`'- foo/*/bar'` Exclut tout fichier nommé `bar` 2 ou plusieurs niveaux sous un répertoire nommé `foo`

`'+ */', '+ *.c', '- *'` La combinaison de ces motifs incluent tous les répertoires et les fichiers source C mais rien d'autre (voir également l'option `-prune-empty-dirs`)

`'+ foo/', '+ foo/bar.c', '- *'` La combinaison de ces motifs incluent seulement le répertoire `foo` et `foo/bar.c`

Les modifieurs suivants sont acceptés après un `'+'` ou `'-'` :

- un `'/'` spécifie que la règle d'inclusion/exclusion doit matcher le chemin absolu de l'élément. Par exemple, `'-/etc/password'` exclut le fichier `/etc/passwd`, et `'-/subdir/foo'` exclut toujours `foo` quand il est dans un répertoire nommé `subdir`.
- un `'!` spécifie que l'inclusion/exclusion prend effet si le motif échoue au matche. Par exemple, `'- ! */'` exclut tous les non-répertoires
- un `'C'` est utilisé pour indiquer que toutes les règles d'exclusion CVS globales devraient être insérées comme exclusion à la place de `'-C'`. n'a pas d'argument
- un `'s'` est utilisé pour indiquer que la règle s'applique côté émetteur. Quand une règle affecte l'émetteur, il empêche les fichiers d'être transférés. Par défaut, un règle affecte les 2 côté sauf avec `-delete-excluded`, auquel cas seul l'émetteur est affecté.
- un `'r'` est utilisé pour indiquer que la règle s'applique au receveur. Quand une règle affecte le receveur, il empêche la suppression des fichiers.
- un `'p'` indique qu'une règle est périssable, signifiant qu'elle est ignorée dans les répertoires qui sont supprimés. Par exemple, les règles `-C` qui excluent des éléments tels que CVS et `*.o` sont marqués comme périssable, et n'empêchent pas un répertoire qui a été supprimé dans la source d'être supprimé dans la destination

## Règles de filtre merge-file

Il est possible de fusionner tous les fichiers dans des règles de filtrage en spécifiant soit une règle de filtre merge (`.`), ou un dir-merge (`:`). Un fichier est est lus une seule fois, et ses règles sont incorporées dans la liste de filtre à la place de la règle `'.'`. pour les fichiers merge par répertoire, `rsync` scanne chaque répertoire qu'il traverse à la recherche du fichier, et fusionne son contenu dans la liste courante des règles héritées. Ces règles par répertoire doivent être créés côté émetteur.

Par exemple :

```
merge /etc/rsync/default.rules
. /etc/rsync/default.rules
dir-merge .per-dir-filter
dir-merge,n- .non-inherited-per-dir-excludes
:n- .non-inherited-per-dir-excludes
```

---

Les modifieurs suivants sont acceptés après une règle merge ou dir-merge :

- un '-' spécifie que le fichier devrait consister seulement de motifs d'exclusion
- un '+' spécifie que le fichier devrait consister seulement de motifs d'inclusion
- un 'C' est une manière de spécifier que le fichier devrait être lu à la manière d'un CSV. Cela active 'n', 'w', et '-', mais autorise également le jeton '!'.
  - un 'e' exclut le nom du merge-file du transfert
  - un 'n' spécifie que les règles ne sont pas héritées par les sous-répertoires
  - un 'w' spécifie que les règles sont séparés par un espace blanc au lieu d'être séparés par des newlines. Cela désactive les commentaires

Les règles sont héritées sauf si 'n' est spécifié. Chaque règle de sous-répertoire a une priorité supérieure aux règles héritées. Quand la règle d'effacement de liste est lu dans un fichier par-répertoire, il efface seulement les règles héritées. Une autre manière d'empêcher l'héritage d'une règle est de l'attacher avec un '/'. Ces règles sont relatives au répertoire du merge-file, dont un motif '/foo' ne match que le fichier foo dans le répertoire.

Exemple de fichier de filtre qui peut être spécifié via `--filter="file"`

```
merge /home/user/.global-filter
- *.gz
dir-merge .rules
+ *. [ch]
- *.o
```

Cela fusionne le contenu du fichier `/home/user/.global-filter` au début de la liste et active le fichier `.rules` comme filtre par répertoire. Toutes les règles lues avant le début du scan de répertoire suivent la règle globales.

Si un merge-file par répertoire est spécifié avec un chemin qui est un répertoire parent du premier répertoire, rsync scan tous les répertoires parents depuis le point de départ à la recherche du fichier par-répertoire. Par exemple :

```
-filter=': /.rsync-filter'
```

Indique à rsync de scanner ce fichier dans tous les répertoire depuis la racine vers le répertoire parent du transfert avant de commencer le scan de répertoire normal. Quelques exemples de ce pré-scan :

```
rsync -avF /src/path/ /dest/dir
rsync -av -filter=': ../../.rsync-filter' /src/path/ /dest/dir
rsync -av -filter=': .rsync-filter' /src/path/ /dest/dir
```

Les 2 premières commandes recherchent `.rsync-filter` dans '/' et '/src' avant que le scan normal commence à regarder dans `/src/path` et ses sous-répertoires. La dernière commande évite le scan de répertoire parent et ne recherche que les fichiers `.rsync-filter` dans chaque répertoire qui font partie du transfert.

Pour inclure le contenu d'un `.cvsignore` dans les motifs, il faut utiliser la règle `:C`, qui crée un dir-merge de fichier `.cvsignore`, mais parsé d'une manière compatible cvs. On peut l'utiliser pour affecter, quand l'inclusion de l'option `-C` du fichier `.cvsignore` par-répertoire est placé dans les règles avec `:C`. Sans cela, rsync ajoute la règle dir-merge pour le fichier `.cvsignore` à la fin de toutes les autres règles. Par exemple :

```
cat «EOT | rsync -avC -filter='.' -' a/ b
+ foo.o
:C
- *.old
EOT
rsync -avC -include=foo.o -f :C -exclude='*.old' a/ b
```

Ces commandes sont identiques. Chacune va fusionner les règles `.cvsignore` par-répertoire au milieu de la liste au lieu de la fin. Cela permet aux règles par répertoire de remplacer les règles qui suivent le `:C` au lieu d'être subordonné à toutes les règles. Pour affecter les autres règles CVS d'exclusion (par ex. la liste d'exclusion par défaut, le contenu de `$HOME/.cvsignore`, et la valeur de `$CVSIGNORE`) il faut omettre `-C` et insérer un règle `-C` dans les règles de filtre ; par ex : `--filter=-C`



---

# Règle de filtre d'effacement le liste

Pour effacer la liste d'inclusion/exclusion, utiliser la règle '!'. La liste courante est soit la liste globale de règle (si la règle est rencontrée en parcourant les options de filtre), ou un jeu de règles par répertoire (qui sont hérités dans leur propre sous-liste, donc un répertoire peut l'utiliser pour effacer les règles parent)

## Motifs include/exclude attachés

Comme mentionné plus haut, les motifs include/exclude globaux sont attachés à la racine du transfert. Cette racine gouverne les motifs commençant par un '/'. Parce que le matching est relatif à cette racine, changer le slash dans le chemin source ou changer l'utilisation de l'option `-relative` affecte le chemin à utiliser dans les matches. Les exemples suivants le démontrent.

Disons que l'on souhaite matcher 2 fichiers sources, un avec un chemin absolu `"/home/me/foo/bar"` et un avec un chemin `"/home/you/bar/baz"`. Voici divers choix de commande qui diffèrent pour un transfert :

```
#> rsync -a /home/me /home/you /dest
+/- pattern: /me/foo/bar
+/- pattern: /you/bar/baz
Target file: /dest/me/foo/bar
Target file: /dest/you/bar/baz
```

```
#> rsync -a /home/me/ /home/you/ /dest
+/- pattern: /foo/bar (note missing "me")
+/- pattern: /bar/baz (note missing "you")
Target file: /dest/foo/bar
Target file: /dest/bar/baz
```

```
#> rsync -a -relative /home/me/ /home/you /dest
+/- pattern: /home/me/foo/bar (note full path)
+/- pattern: /home/you/bar/baz (ditto)
Target file: /dest/home/me/foo/bar
Target file: /dest/home/you/bar/baz
```

```
#> cd /home; rsync -a -relative me/foo you/ /dest
+/- pattern: /me/foo/bar (starts at specified path)
+/- pattern: /you/bar/baz (ditto)
Target file: /dest/me/foo/bar
Target file: /dest/you/bar/baz
```

## Règles par-répertoire et suppression

Sans option `delete`, les règles par-répertoire sont seulement significatives du côté de l'émetteur, donc on peut exclure les fichiers merge eux-mêmes sans affecter le transfert. Pour simplifier, le modifier 'e' ajoute cette exclusion :

```
rsync -av -filter=': .excl' -exclude=.excl host:src/dir /dest
rsync -av -filter=':e .excl' host:src/dir /dest
```

Cependant, pour faire une suppression dans l'émetteur et exclure certains fichiers de la suppression, il faut s'assurer que le receveur connaisse les fichiers à exclure. La manière la plus simple est d'inclure les fichiers merge dans le transfert et utiliser `-delete-after`, parce que cela s'assure que le receveur a toutes les règles d'exclusion avant de tenter quoi que ce soit :

```
rsync -avF -delete-after host:src/dir /dest
```

---

Cependant, si les fichiers merge ne font pas partie du transfert, il faut spécifier des règles d'exclusion globales, ou maintenir les fichiers merge par répertoire côté receveur. Par exemple :

```
rsync -av -filter=': .rules' -filter='. /my/extra.rules' -delete host:src/dir /dest
```

Le fichier extra.rules peut affecter les 2 côtés du transfert, mais côté emeteur les règles sont subordonnés aux règles fusionnées depuis les fichiers .rules. Dans un dernier exemple, le distant exclus les fichiers .rsync-filter du transfert, mais on souhaite utiliser les fichiers .rsync-filter pour contrôler ce qui est supprimé. Pour cela il faut spécifiquement exclure les fichiers de merge par répertoire et placer les règles dans les fichiers locaux pour contrôler ce qui ne devrait jamais être supprimé :

```
rsync -av -filter=':e /.rsync-filter' -delete host:src/dir /dest
```

```
rsync -avFF -delete host:src/dir /dest
```

## Mode batch

Le mode batch peut être utilisé pour appliquer le même jeu de mises à jours à de nombreux systèmes identiques. Supposons une arborescence qui est répliquée dans plusieurs hôtes. Supposons que certains changements ont été fait dans cette arborescence et doivent être propagés sur tous les hôtes. Pour faire celà en mode batch, rsync est lancé avec l'option write-batch, qui stocke toutes les informations dans un fichier.

En générant le fichier batch une fois effectue le status de fichier, checksum et génération de block de données. Les protocoles de transport multicast peuvent être utilisés pour transférer les fichiers de mises à jours batch en parallèle.

Pour appliquer les changements enregistrés dans une autre arborescence de destination, lancer rsync avec l'option read-batch, en spécifiant le nom du fichier et de l'arborescence. Un fichier script peut également être créé quand l'option write-batch est utilisé : il est nommé avec le même nom que le fichier batch avec l'extension '.sh'. Ce script contient une ligne de commande utilisable pour mettre à jour une arborescence de destination en utilisant le fichier batch associé.

### Exemples :

```
$ rsync -write-batch=foo -a host:/source/dir/ /adest/dir/
$ scp foo* remote:
$ ssh remote ./foo.sh /bdest/dir/
```

```
$ rsync -write-batch=foo -a /source/dir/ /adest/dir/
$ ssh remote rsync -read-batch=- -a /bdest/dir/ <foo
```

Dans ces exemples, rsync est utilisé pour mettre à jours /adest/dir/ depuis /source/dir/ et les information sont stockés dans 'foo' et 'foo.sh'. L'hôte distant peut ainsi être mis à jours avec les données batchés pour le répertoire /bdest/dir. La différence entre les 2 exemples révèlent une certaine flexibilité sur la manière de gérer les batches :

- Le premier exemple montre que la copie initiale n'a pas à être locale, on peut pousser ou envoyer les données depuis/vers un hôte distant en utilisant un shell distant ou le service rsync
- Le premier exemple utiliser le fichier foo.sh pour obtenir les options rsync en lançant la commande read-batch sur l'hôte distant.
- Le second exemple lit le batch via l'entrée standard donc le batch n'a pas à être copié dans la machine distante. Cet exemple évite le fichier foo.sh.

L'option read-batch s'attend à ce que l'arborescence de destination qui est mise à jours soit identique à l'arborescence de destination utilisée pour créer le batch. Quand une différence est trouvée la mise à jour n'est pas effectuée avec une alerte (si le fichier apparaît être déjà à jour) ou tenté et donc, si le fichier échoue la vérification, la mise à jour n'est pas effectuée avec une erreur. Cela signifie que relancer l'opération est sûre si la commande est interrompue. Pour forcer la mise à jour à toujours tenter sans regarder la taille et les données, utiliser l'option -I. Si une erreur se produit, l'arborescence de destination va probablement être dans un état de mise à jour partielle. Dans ce cas, rsync peut être utilisé dans son mode régulier non-batch pour fixer la destination.

La version rsync utilisé dans toutes les destinations doit être au moins aussi récente que celle qui a été utilisée pour générer le batch. rsync s'arrête si la version du protocole dans le batch est trop récent.

---

En lisant un batch, rsync force la valeur de certaines options pour matcher les données dans le fichier batch si elles ne sont pas définies à la même valeur que la commande write-batch. D'autres options peuvent être changées. Par exemple, `-write-batch` changé en `-read-batch`, `-files-from` est supprimé, et les options `-filter/-include/-exclude` ne sont pas nécessaires à moins qu'une option `-delete` soit spécifiée

Le code qui crée le batch.sh transforme toutes les options filter/include/exclude en une simple liste qui est ajoutée comme un "here" document dans le script shell. Tout utilisateur avancé peut l'utiliser pour modifier la liste si un changement est désiré avec `-delete`.

## Liens symboliques

3 comportements sont possible quand rsync rencontre un lien symbolique dans le répertoire source. Par défaut les liens symbolique ne sont pas transférés du tout. Un message "skipping non-regular" est émis. Si `-links` est spécifié, les symlinks sont recréés avec la même cible. Si `-copy-links` est spécifié, leur référent est copié au lieu du lien. rsync peut également distinguer les liens sûres et non-sûres. Par exemple, cela peut être utilisé pour un site web miroir pour s'assurer que le module rsync qui est copié n'inclus pas les liens symboliques vers des fichier qui pointent sur la destination. `-safe-links` omet les liens non-sûres.

Les liens symboliques sont considérés comme non-sûres s'ils sont des liens absolus (commençant par '/'), vides, ou s'ils contiennent suffisamment de '..'. Voici un sommaire des options de liens et leur interprétation. Cette liste est dans l'ordre de précedence :

- copy-links** Transforme tous les liens en fichiers normaux
- links, -copy-unsafe-links** Transforme tous les liens non-sûres en fichier et duplique tous les liens sûrs.
- copy-unsafe-links** Transforme tous les liens non-sûres en fichier, et ignore les liens sûres
- links, -safe-links** Duplique les liens sûres et ignore les non-sûres
- links** Duplique tous les liens

## Diagnostic

rsync produit occasionnellement des messages d'erreur qui peuvent être difficile à comprendre, tel que "protocol version mismatch - is your shell clean?". Ce message est généralement généré par les scripts de démarrage ou shell distant produisant des données incorrectes dans le flux que rsync utilise pour son transport. Pour diagnostiquer ce problème, lancer un shell distant comme :

```
ssh remotehost /bin/true > out.dat
```

et lire ce fichier. Si tout est correcte, il devrait être vide.

## Codes de sortie

- 0** succès
- 1** Erreur de syntaxe ou d'utilisation
- 2** Incompatibilité de protocole
- 3** Erreurs en sélectionnant les fichiers/répertoires d'entrée/sortie
- 4** Action demandée non supportées
- 5** Erreur en démarrage le protocole
- 6** Service incapable d'ajouter dans le fichier de log
- 10** Erreur d'E/S socket
- 11** Erreur d'E/S dans le fichier
- 12** Erreurs avec les diagnostics
- 14** Erreur dans le code IPC
- 20** SIGUSR1 ou SIGINT reçu

- 
- 21 Erreurs retournées par waitpid()
  - 22 Erreur en allouant les buffers mémoire
  - 23 Transfert partiel à cause d'erreur
  - 24 Transfert partiel à cause de fichiers sources disparus
  - 25 Limite `-max-delete` atteint
  - 30 Timeout dans l'envoi/reception des données
  - 35 Timeout durant la connexion avec le service

## Variables d'environnement

**CVSIGNORE** Définit les fichiers `.cvsignore`

**RSYNC\_ICONV** Spécifie le paramètre `-iconv`

**RSYNC\_PROTECT\_ARGS** Spécifie une valeur non-zéro pour activer l'option `-protect-args`

**RSYNC\_RSH** Définit le shell par défaut utilisé comme transport pour rsync

**RSYNC\_PROXY** Redirige le client rsync pour utiliser un proxy web

**RSYNC\_PASSWORD** Mot de passe pour lancer une connexion authentifiée

**USER** nom de l'utilisateur

**LOGNAME** Nom de l'utilisateur

**HOME** Répertoire personnel de l'utilisateur pour trouver le fichier `.cvsignore`