

---

# multipath.conf

## Fichier de configuration pour multipath

Chaque section contient un ou plusieurs attributs ou sous-sections. Les mots clé reconnus pour les attributs ou sous-sections dépendent de la section dans laquelle elles se trouvent. Les sections suivantes sont reconnues :

- defaults** Cette section définit les valeurs par défaut pour les attributs qui sont utilisés quand aucune valeur n'est donnée dans le périphérique approprié ou de nom de section
- blacklist** Définit quels périphériques devraient être exclus de la découverte de topologie multipath
- blacklist\_exceptions** Définit quels périphériques devraient être inclus dans la découverte de topologie multipath, même s'ils sont listés dans la section blacklist
- multipaths** Définit les topologies multipath. Elles sont indexées par un WWID (Word Wide Identifier)
- devices** Définit les paramètres spécifiques à un périphérique
- overrides** Définit les valeurs pour les attributs qui devraient remplacer les paramètres spécifiques au périphérique pour tous les périphériques

## Section defaults

- verbosity** Niveau de verbosité. (de 0 à 6, défaut : 2)
- polling\_interval** Intervalle entre 2 vérifications de chemin, en seconde. Pour un fonctionnement correcte, l'intervalle entre les vérifications augmente graduellement jusqu'à `max_polling_interval`. Cette valeur sera remplacée par le paramètre `WatchdogSec` dans `multipathd.service` si `systemd` est utilisé. (défaut : 5).
- max\_polling\_interval** Intervalle maximum entre 2 vérifications de chemin, en secondes. Défaut : `4 * polling_interval`
- reassign\_maps** Active le réassignement des mappages device-mapper. Avec cette option, `multipathd` remap les mappages DM pour continuellement pointer vers un périphérique multiple, et non les périphériques block sous-jacent. Défaut : `no`
- multipath\_dir** Répertoire où les objets partagés sont stockés. Défaut : `/lib[64]/multipath/`
- path\_selector** Chemin par défaut pour l'algorithme à utiliser. Généralement offert par le kernel :
  - round-robin 0** Boucle tout chemin dans le groupe de chemin, envoyant la même quantité d'E/S dans chaque
  - queue-length 0** Envoie le prochain pool d'E/S sur le chemin ayant le moins de traitement E/S en cours
  - service-time 0** Choisi le chemin pour le prochain pool d'E/S basé sur la quantité de traitement d'E/S dans le chemin et sa bande passante relative.
- path\_grouping\_policy** Stratégie de groupement de chemin par défaut à appliquer aux multipaths non-spécifiés :
  - failover** Un chemin par groupe de priorité
  - multibus** Tous les chemins dans un groupe de priorité
  - group\_by\_serial** Un groupe de priorité par numéro de série
  - group\_by\_prio** Un groupe de priorité par valeur de priorité.
  - group\_by\_node\_name** Un groupe de priorité par nom de nœud cible
- uid\_attribute** Attribut `udev` fournissant un identifiant unique de chemin
- prio** Le nom de la routine de priorité de chemin. La routine spécifiée devrait retourner une valeur numérique spécifiant la priorité relative de ce chemin. Actuellement, les routines suivantes sont implémentées :
  - const** Retourne une priorité constante de 1
  - sysfs** Utilise les attributs `sysfs access_state` et `preferred_path` pour générer la priorité de chemin. Accepte l'option `exclusive_pref_bit`

---

**emc** dépendant du hardware. Génère la priorité de chemin pour les array de classe DGC

**alua** dépendant du hardware. Génère la priorité de chemin basé sur les paramètres SCSI-3 ALUA

**ontap** dépendant du hardware. Génère la priorité de chemin pour la classe RDAC LSI/Engenio/NetApp

**hp\_sw** dépendant du hardware. Génère la priorité de chemin pour les arrays HP/COMPAQ/DEC HSG80 et MSA/HSV

**hds** dépendant du hardware. Génère la priorité de chemin pour les arrays Hitachi HDS Modular

**random** Génération aléatoire entre 1 et 10

**weightedpath** Génère la priorité de chemin basé sur l'expression régulière et la priorité fournie come argument nécessite prio\_args

**prio\_args** Arguments à passer à la fonction prio

**features** Spécifie des fonctionnalités device-mapper à utiliser. La syntaxe est **num list** ou num est le numéro, entre 0 et 6, des fonctionnalités dans list. Les valeurs possible pour liste sont :

**queue\_if\_no\_path** Met en queue si aucun chemin n'est actif

**no\_partitions** Désactive la génération de partitions automatique via kpartx

**pg\_init\_retries** Nombre de tentatives pg\_init, doit être entre 1 et 50

**pg\_init\_delay\_msecs** Nombre de secondes avant une tentative pg\_init, doit être entre 0 et 60000

**path\_checker** Méthode par défaut à utiliser pour déterminer les états de chemin. Les valeurs possibles sont :

**tur** Émet une commande TEST UNIT READY au périphérique

**emc\_clarification** dépendant du hardware. requête la page spécifique EVPD 0xC0 DGC/EMC pour déterminer l'état de chemin pour les arrays CLARiiON CX/AX et EMC VNX

**hp\_sw** dépendant du hardware. Vérifie l'état de chemin pour les arrays HP/COMPAQ/DEC HSG80 et MSA/HSV

**rdac** dépendant du hardware. Vérifie l'état de chemin pour la classe RDAC LSI/Engenio/NetApp

**cciss\_tur** dépendant du hardware. Vérifie l'état de chemin pour les contrôleurs HP/COMPAQ Smart Array(CCISS)

**alias\_prefix** Préfix user\_friendly. Défaut : mpath

**failback** Indique comment gérer les erreurs de groupe de chemin :

**immediate** failback immédiatement au chemin du groupe à la priorité la plus élevée qui contient des chemins actifs.

**manual** N'effectue pas de failback automatique

**followover** N'effectue un failback automatique quand quand le premier chemin d'un pathgroup devient actif. Cela permet d'empêcher un nœud d'être rejeté quand un autre nœud demande le failover

**values > 0** Défère le failback (temps en secondes)

**rr\_min\_io** Nombre d'E/S à route dans le chemin avant de basculer dans le suivant dans le même groupe de chemin. Uniquement pour le multipath basé sur BIO. Défaut : 1000

**rr\_min\_io\_rq** Nombre de requêtes d'E/S à router vers un chemin avant de basculer vers le suivant dans le même groupe de chemin. Uniquement pour les requêtes basées sur multipath. Défaut : 1

**max\_fds** Nombre maximum de descripteurs de fichier qui peuvent être ouverts par multipath et multipathd. 'max' prend la valeur dans /proc/sys/fs/nr\_open. Si non définis, utilise open\_fds du processus appelant. généralement 1024. Pour plus de sécurité devrait être le nombre max de chemins + 32, si ce nombre est supérieur à 1024. Défaut : max

**rr\_weight** à priorities, le configurateur multipath assigne un poids de chemin tel que "path prio \* rr\_min\_io". peut être priorities ou uniform

**no\_path\_retry** Spécifie le nombre de tentatives jusqu'à désactiver la queue, ou fail pour échouer immédiatement. queue ne stop jamais la queue. Si non définis, aucune queue n'est tentée

**queue\_without\_daemon** à no, quand multipathd stoppe, la queue est désactivée pour tous les périphériques. Utile pour les périphériques qui définissent no\_path\_retry. Si une machine s'éteint alors que tous les chemins sont down, il est possible de bloquer en attente d'E/S du périphérique après que multipathd ai été stoppé. Sans multipathd, l'accès aux chemins ne peut être restauré, et le kernel ne peut pas indiquer de stopper la queue. Défaut : no

**checker\_timeout** Spécifie le timeout à utiliser pour la vérification de chemin et es prioritiseurs qui émettent des commandes SCSI avec un timeout explicite. Défaut : /sys/block/sd<x>/device/timeout

**flush\_on\_last\_del** À yes, multipathd désactive la queue quand le dernier chemin vers un périphérique a été supprimé. Défaut : no

**user\_friendly\_names** À yes, en utilisant le fichier /etc/multipath/bindings pour assigner de manière persistante et unique un alias au multipath, sous la forme mpath<n>. À no, utilise le wwid comme alias.

---

**fast\_io\_fail\_tmo** Nombre de secondes d'attente de la couche SCSI après qu'un problème ait été détecté sur un port distant FO avant d'échouer les E/S sur ce port. Devrait être inférieur à `dev_loss_tmo`. Défaut : 5

**dev\_loss\_tmo** Nombre de secondes d'attente de la couche SCSI après qu'un problème ait été détecté sur un port distant FO avant de supprimer du système. 'infinity' définis à la valeur max (2147483647 secondes, ou 68ans). Il sera automatiquement ajusté à l'intervall `no_path_retry` addentry articles autoadd autofind autoprod createalpha createbeta createdb createprod findentry fullpowa generator.php genhtml genman genmd gentex html insert man md pdf regen setfor setfor2 sql temp temp-new-pc tex threads ToDo polling\_interval si un nombre de tentatives est donné dans `no_path_retry` et que l'intervall de retentative qlobal est supérieur à `dev_loss_tmo`. Linux définis cette valeur à 300 si `fast_io_fail_tmo` n'est pas définis. Défaut : 600

**bindings\_file** Chemin complet du fichier binding à utiliser quand l'option `user_friendly_names` est définis.

**wwids\_file** Chemin complet du fichier eds WWID, utilisé pour garder une trace des WWID pour les LUN qui ont été créés dans le passé

**log\_checker\_err** À 'once', multipathd log la première erreur de vérification de chemin au niveau 2, et les autres erreur au niveau 3 jusqu'à ce que le périphérique soit restauré. À 'always', multipathd logs toujours au niveau 2

**reservation\_key** Clé de réservation d'action du service utilisé par mpathpersist. Il doit être définis pour tous les périphériques multipath utilisant les réservation persistantes, et doit être identique au champ RESERVATION KEY du paramètre PERSISTENT RESERVE OUT.

**retain\_attached\_hw\_handler** À yes, si la couche SCSI a déjà attaché un hardware\_handler au périphérique multipath, multipath ne force pas le périphérique à utiliser celui spécifié dans ce fichier de configuration. Défaut : yes

**detect\_prio** À yes, multipath tente de détecter si le périphérique supporte SCSI-3 ALUA. Si c'est le cas, le périphérique utilisera automatiquement le prioritizer sysfs si les attributs `access_state` et `preferred_path` sont supportés, ou le prioriseur alua le cas échéant.

**force\_sync** À yes, multipathd Vérifie les chemin en mode sync uniquement. Cela signifie que seul un vérificateur tourne à la fois, utile quand les vérificateurs tournant en parallèle créent une surcharge CPU.

**strict\_timinig** À yes, multipathd démarre une nouvelle boucle de vérification de chemin après exactement 1 seconde, pour que chaque vérification de chemin se produise exactement à `polling_interval` secondes. Si les systèmes surchargés, les vérification peuvent prendre plus d'une seconde

**deferred\_remove** À yes, multipathd déferre la suppression au lieu d'une suppression régulière quand le dernier chemin a été supprimé. Cela signifie que si le périphérique continue à être utilisé, il sera libéré quand le dernier utilisateur le fermera.

**partition\_delimiter** Non définis, quand multipath renomme un périphérique, il agit simplement comme `kpartx`. ajoute seulement un 'p' aux noms se terminant dans un nombre. Si définis, multipath agit comme "`kpartx -p`" et ajoute toujours un délimiteur

**config\_dir** Si définis, multipath recherche dans ce répertoire pour des fichiers .conf, et y lit les informations de configuration, tout comme `/etc/multipath.conf`.

**delay\_watch\_checks** > 0, multipathd recherche les chemins qui sont devenus récemment valides durant ce délai.

**delay\_wait\_checks** > 0, quand un périphérique qui est récemment revenu online échoue de nouveau dans le délai `delay_watch_checks`, la prochaine fois qu'il revient online il sera marqué et retardé, et ne sera pas utilisé jusqu'à ce qu'il ait passé ces vérification

**find\_multipaths** À yes, au lieu de tenter de créer un périphérique multipath pour chaque chemin non-blacklisté, multipath créé seulement un périphérique si une des 3 conditions sont rencontrées : 1) il y a au moins 2 chemins non-blacklistés avec le même WWID, 2) l'utilisateur force manuellement la création en spécifiant un périphérique avec la commande `multipath`, ou 3) un chemin a le même wwid précédemment créé. Défaut : no

**uxsock\_timeout** timeout CLI reçu en ms. Défaut : 1000

**retrigger\_tries** Nombre de fois que multipathd tente de gérer un uevent pour obtenir le WWID. Défaut : 3

**retrigger\_delay** Définis le délai, en secondes, d'attente entre les triggers. Défaut : 10

**missing\_uev\_wait\_timeout** Contrôle le délai d'attente de multipathd après qu'un nouveau périphérique multipath soit créé, pour recevoir un évènement de changement de udev pour ce périphérique, avant d'activer automatiquement les reloads de périphérique. Défaut : 30

## Section blacklist

**devnode** expression régulière des périphériques à exclure

**wwid** Le wwid d'un périphérique

---

**property** Expression régulière d'une propriété udev à exclure

**device** Sous-section pour la description d'un périphérique. Accepte les mots-clé vendor et product

## Section blacklist

**devnode** expression régulière des périphériques à inclure

**wwid** Le wwid d'un périphérique

**property** Expression régulière d'une propriété udev à inclure

**device** Sous-section pour la description d'un périphérique. Accepte les mots-clé vendor et product

## Section multipaths

**multipath** Sous-section décrivant un multipath

**wwid** Index du conteneur

**alias** nom symbolique pour le mappage multipath

**path\_grouping\_policy**

**path\_selector**

**prio**

**prio\_args**

**failback**

**rr\_weight**

**no\_path\_retry**

**rr\_min\_io**

**rr\_min\_io\_rq**

**flush\_on\_last\_del**

**features**

**reservation\_key**

**user\_friendly\_names**

**deferred\_remove**

**delay\_watch\_checks**

**delay\_wait\_checks** Identiques à la section defaults

## Section devices

**device** Sous-section décrivant un périphérique

**vendor** Identifiant de vendeur

**product** Identifiant de produit

**revision** Identifiant de révision

**product\_blacklist** Chaîne à blacklister pour ce vendeur

**alias\_prefix** Préfix user\_friendly à utiliser pour ce type de périphérique

**hardware\_handler** hardware handler à utiliser pour ce type de périphérique :

**emc** DGC class arrays as CLARiiON CX/AX and EMC VNX

---

**rdac** LSI/Engenio/NetApp RDAC class  
**hp\_sw** HP/COMPAQ/DEC HSG80 and MSA/HSV arrays  
**alua** SCSI-3 ALUA compatible arrays

**path\_grouping\_policy**

**uid\_attribute**

**path\_selector**

**path\_checker**

**prio**

**prio\_args**

**features**

**failback**

**rr\_weight**

**no\_path\_retry**

**rr\_min\_io**

**rr\_min\_io\_rq**

**fast\_io\_fail\_tmo**

**dev\_loss\_tmo**

**flush\_on\_last\_del**

**retain\_attached\_hw\_handler**

**detect\_prio**

**deferred\_remove**

**delay\_watch\_checks**

**delay\_wait\_checks** Options identiques à la section defaults

## Section overrides

**path\_grouping\_policy**

**uid\_attribute**

**getuid\_callout**

**path\_selector**

**path\_checker**

**alias\_prefix**

**features**

**prio**

**prio\_args**

**failback**

**rr\_weight**

**no\_path\_retry**

**rr\_min\_io**

**rr\_min\_io\_rq**

**flush\_on\_last\_del**

**fast\_io\_fail\_tmo**

**dev\_loss\_tmo**

**user\_friendly\_names**

**retain\_attached\_hw\_handler**

**detect\_prio**

**deferred\_remove**

**delay\_watch\_checks**

**delay\_wait\_checks** Options identiques à la section devices ou defaults