

Manuel de l'utilisateur Inotify-daemon

Inotify-daemon, version 0.1.0, 25/08/2022

Christian CAMIER christian.c@promethee.services

Copyright © 2022 Promethee Services Copyright © 2022 Christian CAMIER <christian.c@promethee.services>

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Short Contents

1	Introduction	1
2	Lancer inotify-daemon	2
3	Configuration d'inotify-daemon	4
4	Gestion des journaux systèmes (logs)	11

1 Introduction

1.1 Description

Inotify-daemon est un démon qui permet, sur des répertoires donnés, d'initier des actions lorsque certains événements du système de fichiers surviennent.

2 Lancer inotify-daemon

2.1 Préséance des arguments en ligne sur la configuration

Lorsque Inotify-daemon est invoqué, les arguments en ligne de commande ont priorité sur les directives correspondantes donnés dans le fichier de configuration.

2.2 Options en lignes

2.2.1 Les formats des options en ligne

2.2.1.1 Options courtes et options longues

Inotify-daemon propose deux styles d'options :

- Les options courtes, précédées par un unique signe moins (-), composées d'un unique caractère.
- Les options longues, précédées par deux signe moins (--) et composées de plusieurs caractères pour être plus parlantes.

2.2.1.2 Les options courtes

Les options nécessitant un argument, ce dernier est placé dans l'argument qui suit l'option.

Exemple :

```
inotify-daemon -C /home/foo/myconfiguration.cfg
```

Plusieurs options peuvent être cumulées sur le même argument de la commande en ligne. Ainsi, les deux formes suivantes sont équivalentes :

```
inotify-daemon -f -T
inotify -fT
```

Lorsque plusieurs options sont cumulées et que certaines d'entre elles nécessite un argument, les arguments de la ligne de commande suivants, dans l'ordre d'apparition des options seront utilisés. Les deux formes suivantes sont donc équivalentes :

```
inotify-daemon -CfLl /tmp/foo.cfg syslog info
inotify-daemon -C /tmp/foo.cfg -f -L syslog -l info
```

2.2.1.3 Les options longues

Les options longues, contrairement aux options courtes, ne peuvent pas être cumulés.

Si une option nécessite un argument, l'argument est accolé à la suite de l'option, dans le même argument de la commande en ligne, séparé par un signe égal (=).

Exemple :

```
inotify-daemon --configuration=/home/foo/myconfiguration.cfg
```

2.2.2 les options

Les options prises en compte par **inotify-daemon** sont les suivantes :

		Description
court C fic	long configuration=fic	Défini le fichier devant être pris en compte au démarrage. L'argument est le chemin d'accès au fichier de configuration désiré. La valeur par défaut est /etc/inotify-daemon.conf .
f	foreground	Exécution en avant plan. Util pour de débogage.
h	help	Affiche une aide succincte de inotify-daemon .
L drv	logdriver=drv	Défini le pilote devant être utilisé pour la consignation des traces d'exécution. Pour plus d'information, voir la partie "Gestion des journaux systèmes(logs)".
l lvl	loglevel=lvl	Défini le niveau de trace minimal à consigner dans les journaux systèmes. Pour plus d'information, voir la partie "Gestion des journaux systèmes(logs)".
o o=v	logoption=o=v	Défini une option du système de journalisation. Pour plus d'information, voir la partie "Gestion des journaux systèmes(logs)".
p fil	pidfile=fil	Défini le fichier à utiliser pour stocker l'identifiant du processus (pid) du daemon en cours d'exécution.
R	reload	Demande, au démon en cours d'exécution de relire sa configuration.
S	stop	Demande, au démon en cours d'exécution de s'arrêter.
T	test	Test de la configuration
V	version	Affiche la version du programme.

3 Configuration d'inotify-daemon

3.1 Le fichier de configuration

3.1.1 Présentation

L'ensemble des comportements de **inotify-daemon** sont définis dans un fichier de configuration. Par défaut, le fichier de configuration utilisé est `/etc/inotify-daemon.conf`.

3.1.2 Structure du fichier de configuration

Le fichier de configuration est un fichier texte (ASCII). Les lignes vides (ou remplies uniquement de caractères blancs) et les lignes dont le premier caractère non blanc est un dièse (`#`).

Pour ce qui est des directives, ces dernières sont insensibles à la casse.

Chaque ligne est composée d'une directive, c'est à dire, un mot clé suivi d'éventuels arguments. Certaines de ces directives sont des sections (groupes de directives).

Les arguments sont séparés par un ou plusieurs espaces et/ou tabulations.

Une section commence par une ligne contenant une directive de section et se termine une ligne ne contenant que le mot clé **end**. Les sections peuvent être imbriquées.

Exemples de section simple :

```
directory /var/lib/input
    event close_write fichier_recu.sh %D %F
    ...
end
```

Exemple de sections imbriquées :

```
settings
    ...
    logging file
        set filename /var/log/ind.log
    ...
end
end
```

Les directives de plus haut niveau sont les suivantes :

- settings** Directive unique, normalement en entête du fichier de configuration, initiant une section permettant de définir les comportements globaux d'**inotify-daemon**.
- directory** Directive permettant de définir une section permettant de définir, pour un répertoire donné, les actions à effectuer sur la réception d'événements définis.
- include** Directive permettant d'inclure des fichiers contenant des bouts de configurations.

3.1.3 Les arguments des directives

3.1.3.1 Les différents types d'arguments

Certaines directives prennent des arguments ayant une sémantique spécifique. Ces types d'arguments sont les suivants :

- Les booleens
- Les entiers
- Les valeurs prédéfinies
- Les combinaisons de valeurs prédéfinies

3.1.3.2 Les arguments booleens

Les arguments de type booleens, utilisés comme des interrupteurs. Les valeurs possibles sont **vrai** et **faux**.

Pour désigner ces valeurs dans le fichier de configuration, les possibilités sont les suivantes :

pour vrai	on true yes
pour faux	off false no

3.1.3.3 Les arguments entiers

Certaines directives exigent qu'un ou plusieurs de ses arguments soit une valeur entière positive. Ces valeurs peuvent être en octal, décimal, ou hexadécimal.

Si la valeur indiquée commence par un zéro (**0**), la valeur sera considérée comme octale.

Si la valeur indiquée commence par **0x**, la valeur sera considérée comme hexadécimale.

Toute autre valeur sera considérée comme décimale.

Exemple, pour la valeur (10)_d, les possibilités sont les suivantes :

décimal	10
octal	012
hexadécimal	0xA ou 0xa

3.1.3.4 Les valeurs prédéfinies

Certaines directives exigent qu'un ou plusieurs de ses arguments soit choisi parmi des valeurs prédéfinies. Dans ce cas, voir la documentation de la directive pour connaître les possibilités.

3.1.3.5 Les combinaisons de valeurs prédéfinies

Certaines directives qui exigent qu'un ou plusieurs de ses arguments soit choisi parmi des valeurs prédéfinies accepte des combinaisons de ces valeurs.

Dans ce cas, les différentes valeurs sont placées les unes à la suite des autres, séparées par une virgule (,) ou par une barre verticale (|) sans espaces ni tabulations.

Exemple dans le cas de la directive "event" :

```
event close_write,close_nowrite action
event close_write|close_nowrite action
```

3.2 La directive "settings"

3.2.1 Présentation

Cette directive indique le début de la section dans laquelle les comportements globaux (hors gestion des événements sur les répertoires) sont définis.

Cette directive ne prend pas d'arguments.

3.2.2 Les sous directives de “settings”

Les sous directives de la section **settings** sont les suivantes :

pidfile	Indique le fichier à utiliser pour consigner l'identifiant processus du démon en cours d'exécution.
logging	Indique le début de la sous section dans laquelle est défini les consignation dans les journaux système.

3.2.3 La directive “pidfile”

Cette directive prend, en unique argument, le chemin absolu du fichier qui devra contenir l'identifiant du processus (**PID**) lorsque le démon s'exécute.

La valeur par défaut est `/var/run/inotify-daemon.pid`

3.2.4 La directive “logging”

3.2.4.1 Avant propos

Cette partie ne documente que les directives. Pour plus de détail sur la gestion des logs See Chapter 4 [Gestion des journaux systèmes], page 11.

3.2.4.2 Présentation

Cette directive permet de définir une section dans laquelle il est indiqué comment **inotify-daemon** consigne les traces applicatives.

Cette directive prend, en argument, le pilote à utiliser pour consigner les logs.

Dans la section **logging** les directives suivantes sont acceptées :

level	Définit le niveau à partir duquel les traces sont consignées
set	Permet de définir une option de la consignation des traces.

3.2.4.3 La directive “level”

Cette directive prend, en argument, le niveau minimal des traces applicatives qui seront consignées.

Les niveaux possibles sont :

- **ALERT**
- **CRIT**
- **DEBUG**
- **EMERG**
- **ERROR**
- **INFO**
- **NOTICE**
- **WARNING**

Exemple :

```
level notice
```

Note :

Si cette directive est utilisée plusieurs fois, c'est la dernière occurrence qui impose le niveau minimal.

3.2.4.4 La directive “set”

Cette directive permet de définir une option du système de consignation des traces. Elle prend deux arguments :

option Le nom de l'option.
valeur La valeur assignée à l'option.

Exemples, dans le cas de la journalisation dans un fichier :

```
set filename /var/log/inotify-daemon.log
set stayopen off
```

Exemples, dans le cas de la journalisation au travers du démon syslog :

```
set facility daemon
set ident toto
set options ndelay|perror|pid
```

3.3 La directive “directory”

3.3.1 Présentation

Cette directive permet d'ouvrir une section dans laquelle sont définies les actions devant être exécutées lorsque des événements sont générés au sein d'un répertoire donné.

3.3.2 Syntaxe

Cette directive prend en argument, le répertoire devant être scruté. Elle s'utilise comme suit:

```
directory nomrepertoire
...
end
```

3.3.3 Les sous directives de “directory”

Les sous directives d'une section **directory** sont les suivantes :

user Utilisateur sous l'identité duquel sont lancées les actions.
group Groupe sous l'identité duquel sont lancées les actions.
shell Shell devant être utilisé pour exécuter les actions.
event Définition d'une action pour un ou plusieurs événements.

3.3.4 La directive “user”

Cette directive permet d'indiquer sous quelle identité utilisateur (**uid**) doivent être exécutées les actions.

Exemple :

```
user www-data
```

La valeur par défaut est “nobody”.

3.3.5 La directive “group”

Cette directive permet d'indiquer sous quelle identité de groupe (**gid**) doivent être exécutées les actions.

Exemple :

```
group www-data
```

La valeur par défaut est “nogroup”.

3.3.6 La directive “shell”

Cette directive indique quel langage de commande (shell) utiliser pour interpréter la commande. Le shell par défaut est “/bin/sh”.

Exemple :

```
shell /bin/csh
```

3.3.7 La directive “event”

3.3.7.1 Présentation

Cette directive permet de définir l'action à exécuter lorsque un ou plusieurs événements surviennent sur le répertoire.

3.3.7.2 Les différents événements

Cette directive accepte les événements suivants de façon unitaire ou en combinaison :

ACCESS	Le fichier à l'origine de l'événement a été accédé.
ATTRIB	Les métadonnées du fichier à l'origine de l'événement (droits, propriété, ...) ont été modifiées.
CLOSE	Équivalent à CLOSE_WRITE CLOSE_NOWRITE
CLOSE_WRITE	Le fichier à l'origine de l'événement vient d'être fermé après avoir été ouvert en écriture.
CLOSE_NOWRITE	Le fichier à l'origine de l'événement vient d'être fermé après avoir été ouvert en lecture seule.
CREATE	Le fichier à l'origine de l'événement vient d'être créé.
DELETE	Le fichier à l'origine de l'événement vient d'être supprimé.
DELETE_SELF	Le répertoire en cours de scrutation vient d'être supprimé.
MODIFY	Le fichier à l'origine de l'événement vient d'être modifié.
MOVE_SELF	Le répertoire en cours de scrutation vient d'être déplacé.
OPEN	Le fichier à l'origine de l'événement vient d'être ouvert.
RENAME	Le fichier à l'origine de l'événement vient de changer de nom.

Note :

Pour des actions différentes sur des événements donnés, plusieurs directives **event** peuvent cohabiter au sein d'une section **directory** tant que les événements gérés sont distincts.

3.3.7.3 Syntaxe

Cette directive s'utilise comme suit :

```
event events action
```

Avec :

events :	Un événement ou une combinaison d'événements.
command :	La commande à exécuter. Cette commande commence au premier caractère non blanc après l'argument events et se termine avec la fin de la ligne. Les espaces et tabulations sont donc acceptées. La commande est une chaîne de caractères ayant des directives de formes. A l'exception du caractère pourcent (%) qui est réservé pour les remplacements, les caractères sont pris en compte tel quel. Le caractère pourcent (%) permet de définir des remplacements en fonction du ou des caractères qui le suivent. Ces remplacements sont les suivants :
%	Le caractère pourcent lui même.
c	"Cookie" associé à l'événement en décimal dans le cas RENAME .
cb	"Cookie" associé à l'événement en binaire sur 32 chiffres dans le cas RENAME .
cd	"Cookie" associé à l'événement en décimal dans le cas RENAME .
co	"Cookie" associé à l'événement en octal sur 11 chiffres dans le cas RENAME .
cx	"Cookie" associé à l'événement en hexadécimal minuscule sur 8 chiffres dans le cas RENAME .
eX	"Cookie" associé à l'événement en hexadécimal majuscule sur 8 chiffres dans le cas RENAME .
D	Le nom du répertoire dans lequel l'événement est survenu.
E	Le nom de l'événement.
e	Le masque de l'événement en décimal.
eb	Le masque de l'événement en binaire sur 32 chiffres.
ed	Le masque de l'événement en décimal.
eo	Le masque de l'événement en octal sur 11 chiffres.
ex	Le masque de l'événement en hexadécimal minuscules sur 8 chiffres.
eX	Le masque de l'événement en hexadécimal majuscules sur 8 chiffres.
N	Le nom du fichier ayant causé l'événement. Dans le cas de l'événement RENAME , c'est le nouveau nom.
O	L'ancien nom du fichier dans le cas de l'événement RENAME .

Note :

Par défaut la sortie standard (stdout) de l'action est consignée dans les journaux systèmes avec le niveau **DEBUG** et la sortie d'erreur (stderr) avec le niveau **NOTICE**.

Exemple

```
event rename echo %D %Cx %F renomme en %O
event close_write ls -l %D/%F
```

3.4 La directive “include”

3.4.1 Présentation

Cette directive permet d'inclure des fichiers contenant des éléments de configuration complémentaires. Cette directive est pratique pour éviter, dans le cas où de nombreux répertoires sont scrutés, évitant ainsi d'avoir à traiter un très long fichier de configuration lorsque l'on doit ajouter, supprimer ou modifier un répertoire scruté.

3.4.2 Utilisation

Cette directive ne prend qu'un unique argument et s'utilise comme suit :

include nomfichier

Ainsi l'inclusion d'un unique fichier se fait comme dans l'exemple suivant :

include /etc/inotify-daemon.conf.local

Il est possible de définir un ensemble de noms de fichiers en utilisant des caractères spéciaux tel que défini dans **glob(7)**. Il est ainsi possible l'ensemble des fichiers se terminant avec l'extension “.conf” dans un répertoire donné :

include /etc/inotify-daemon.d/*.conf

A noter que si le fichier inclus n'existe pas (ou que la génération de nom de fichier ne retourne aucun fichier existant), ce ne sera pas considéré comme une erreur de configuration.

4 Gestion des journaux systèmes (logs)

4.1 Les niveaux d'écriture dans les journaux systèmes

4.1.1 Les niveaux

4.1.1.1 Définition des niveaux

Les différents niveaux d'écriture dans les journaux systèmes sont directement calqués sur ceux proposés par syslog(3).

Ces niveaux sont les suivants :

niveau	description
ALERT	Une action doit être entreprise immédiatement.
CRIT	Condition critique.
DEBUG	Information de débogage.
EMERG	Le système n'est plus utilisable.
ERROR	Condition d'erreur.
INFO	Message d'information.
NOTICE	Condition normale mais significative.
WARNING	Message nécessitant une attention.

4.1.1.2 Hiérarchisation des niveaux

Les niveaux de messages sont hiérarchisés. Cela signifie qu'il y a une relation d'importance entre les différents niveaux.

Cette hiérarchisation de ces niveaux est la suivante :

```

^ (+ élevé)
| EMERG
| ALERT
| CRIT
| ERROR
| WARNING
| NOTICE
| INFO
| DEBUG
| (+ bas)
```

Lorsque le niveau des messages est défini (cf. directive **level** de la section **logging**, seuls les messages d'un niveau supérieur ou égal à ce niveau seront consignés.

4.2 Les pilotes

4.2.1 La notion de pilote

La gestion des traces d'**inotify-daemon** peut consigner les informations selon différentes manières. C'est le rôle des pilotes.

Les différents pilotes disponibles sont les suivants :

default	Ce pilote redirige les messages vers la sortie standard ou la sortie d'erreur du terminal.
file	Ce pilote redirige les messages vers un fichier.
syslog	Ce pilote redirige les messages vers le syslog .

Note :

Chaque pilote possède ses propres options.

4.3 Les différentes options

4.3.1 Les options globales

4.3.1.1 les différentes options globales

Les différentes options globales sont les suivantes :

act_err_lvl	Indique le niveau à utiliser pour consigner les sorties d'erreurs lorsqu'une action sur un événement.
act_out_lvl	Indique le niveau à utiliser pour consigner les sorties standard lorsqu'une action sur un événement.
timestamp	Indique le format de l'horodatage des messages.

4.3.1.2 L'option "act_err_lvl"

L'option globale **act_err_lvl** permet de définir avec quel niveau sont consignées les messages émis sur la sortie d'erreur par les actions égnagées suite à un événement.

Sa valeur par défaut est : NOTICE.

4.3.1.3 L'option "act_out_lvl"

L'option globale **act_out_lvl** permet de définir avec quel niveau sont consignées les messages émis sur la sortie standard par les actions égnagées suite à un événement.

Sa valeur par défaut est : DEBUG.

4.3.1.4 L'option "timestamp"

L'option globale **timestamp** permet de définir le format de l'horodatage des lignes consignées dans le journal.

Cette option prend, en argument, une chaîne de caractères dans le format défini par **strftime(3)**. Le valeur par défaut est : %d/%m/%Y %H:%M:%S.

4.3.2 Les options spécifiques à “default”

La seule option spécifique pour le pilote “default” est **channel**. Elle indique sur quel canal doivent être consignées les écritures du journal. Cette option accepte les valeurs suivantes :

stdout Les informations sont écrites sur la sortie standard.
stderr Les informations sont écrites sur la sortie d’erreur.

La valeur par défaut de cette option est la sortie d’erreur (**stderr**).

4.3.3 Les options spécifiques à “file”

4.3.3.1 Les différentes options

Les différentes options prises en compte par le pilote **file** sont les suivantes :

filename Définit le nom du fichier dans lequel seront consignées les écritures dans le journal.
mode Définit le mode de création du fichier.
stayopen Définit si le fichier doit rester ouvert ou être ouvert puis fermé à chaque écriture.
timestamped Indique si le nom de fichier donné avec l’option **filename** est dans un format horodaté.

4.3.3.2 L’option “filename”

La valeur par défaut de cette option est “/var/log/inotify-daemon-%Y%m”. Cette valeur nécessite que l’option **timestamped** soit activée (cf. ci-dessous).

4.3.3.3 L’option “mode”

Mode de création des fichiers. La valeur est un nombre octal compris entre 000 et 777 qui indique les droits d’accès au fichier créé. Ces droits d’accès sont les mêmes que ceux définis par **chmod(2)**. Le mode de création par défaut des fichiers journaux est **600**, c’est à dire que le propriétaire a les droits d’écriture et lecture et que les membres du groupe et autres utilisateurs n’ont aucun droit.

4.3.3.4 L’option “stayopen”

Indique si le fichier doit rester ouvert entre deux écritures. Les valeurs possibles sont les suivantes. Cette option prend une valeur booléenne.

Dans le cas où cette option est activée et que l’option **timestamped** est positionnée, le nom du fichier est recalculé à chaque trace émise. Si le nom du fichier diffère, du précédent, l’ancien fichier est fermé et le nouveau fichier créé.

4.3.3.5 L’option “timestamped”

Indique si le nom du fichier est horodaté. Cette option prend en argument une valeur booléenne. Dans le cas où cette option est activée, le nom du fichier est calculé en fonction du format **strftime(3)**.

4.3.4 Les options spécifiques à “syslog”

4.3.4.1 Les différentes options

Le pilote **syslog** accepte les options suivantes :

facility	Facilité syslog devant être utilisée par syslog(3) .
ident	Identifiant utilisé pour les traves émises dans le syslog(3) .
options	Options de syslog(3) .

4.3.4.2 L’option “facility”

Cette option permet de définir la facilité syslog qui devra être utilisée pour les messages. Les valeurs de “facility” possibles sont les suivantes :

AUTH	Messages ayant trait à la sécurité out aux autorisations.
AUTHPRIV	Messages ayant trait à la sécurité out aux autorisations. (version privé).
CRON	Messages ayant trait aux démons d’horloge (cron et at).
DAEMON	Messages ayant trait aux démons systèmes sans facilité spécifique.
FTP	Messages ayant trait au démons FTP.
KERN	Messages du noyau Linux.
LOCAL0 à LOCAL7	Messages à usage locaux.
LPR	Messages ayant trait au système d’impression.
MAIL	Messages ayant trait au système de mels.
NEWS	Messages ayant trait au système de news.
SYSLOG	Messages générés de façon interne par syslogd(8) .
USER	Messages génériques au niveau utilisateurs.
UUCP	Messages ayant trait au système UUCP.

4.3.4.3 L’option “ident”

Cette option est permet de définir un identifiant qui sera préfixé à chaque message. Cette option prend, en argument, une chaîne de caractères ne comportant aucun blanc (espaces et tabulations).

4.3.4.4 L’option “options”

Cette option permet de fixer les options de **syslog(3)**. Cette option prend, en argument, soit **none** pour indiquer qu’il n’y a pas d’options, soit une ou une combinaison des options suivantes :

CONS	Les messages sont écrits directement sur la console système s’il y a une erreur lors de l’émission du message au syst\’eme de gestion des journalisation (syslog).
NDELAY	Ouvrir la connexion au système de journalisation immédiatement lors de l’initialisation.
NOWAIT	Ne pas attendre le processus fils s’il est créé lors de l’envoi d’un message.
ODELAY	L’inverse de NDELAY . La connexion au système de journalisation est effectuée que lors de l’émission du premier message.
PERROR	Les messages sont aussi écrits sur la sortie d’erreur (stderr).
PID	L’identifiant processus (PID) de l’émetteur est inclus dans chaque message.